M 2014

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MAC FOR WI-FI-BASED LAND-SEA COMMUNICATIONS

JOÃO MIGUEL DE OLIVEIRA MAGALHÃES

DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
MIEEC – MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES | **2013/2014**

A Dissertação intitulada

"MAC for Wi-Fi-Based Land-Sea Communications"

foi aprovada em provas realizadas em 22-07-2014

o júri

Presidente Professor Doutor Mário Jorge Moreira Leitão
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Manuel Tavares Vieira Cabral
Professor Auxiliar do Departamento de Eletrónica Industrial da Escola de Engenharia da Universidade do Minho

Professor Doutor Manuel Alberto Pereira Ricardo
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - João Miguel de Oliveira Magalhães

Faculdade de Engenharia da Universidade do Porto

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

MAC for Wi-Fi-based Land-Sea Communications

João Miguel de Oliveira Magalhães

Mestrado Integrado em Engenharia Eletrotécnica e Computadores

Supervisor: Manuel Ricardo (PhD)

Co-Supervisor: Rui Campos (PhD)

June, 2014

Abstract

Communications are no longer just wired connected clients or servers. Wireless communications are nowadays equally or more important from the end-users point of view. This dissertation focused on wireless maritime communications, an area still underexplored but with a huge potential.

Several wireless communications solutions are already used in maritime environment, such as satellite communications, which is a quite expensive technology and not affordable by most of maritime users, HF/VHF analogue channels which are used in voice communications, and cellular networks (GSM, 3G, LTE) only available close to the shore; WiMAX based solutions have also been evaluated within some private networks around the world for enabling land-sea communications, representing a proprietary solution with increased implementation costs. Wi-Fi represents a future solution for these environments and have already been tested, demonstrating satisfactory results.

In this dissertation we propose a modified IEEE 802.11 MAC including a dynamic MTU algorithm and a companion RSSI prediction algorithm, in order to improve Wi-Fi maritime communications performance. Experimental results show considerable gains for low signal strength regimes when compared to the standard IEEE 802.11 MAC.

Resumo

As comunicações têm vindo a deixar de ser apenas ligações cabladas entre clientes e servidores. As comunicações móveis são atualmente tão ou mais importantes do ponto de vista dos utilizadores. Esta dissertação focou-se em comunicações marítimos móveis, uma área ainda pouco explorada, mas vista como tendo um grande potencial.

Várias soluções para comunicações móveis são atualmente usadas no ambiente marítimo, nomeadamente comunicações via satélite, que representam uma tecnologia dispendiosa, não sendo suportável pela maioria dos utilizadores marítimos, canais analógicos HF/VHF que são usados para comunicações de voz e redes celulares (GSM, 3G, LTE), estando apenas disponíveis perto da costa; Soluções baseadas em WiMAX têm também vindo a ser avaliadas em algumas redes privadas à escala mundial para implementar comunicações terra-mar, estas soluções representam uma tecnologia de licença proprietária com elevados custos de implementação. O Wi-Fi representa então uma solução que futuramente pode vir a ser usada nestes ambientes, sendo que já tem vindo a ser testada com resultados satisfatórios.

Nesta dissertação propomos uma modificação ao MAC IEEE 802.11, que incluí um algoritmo de MTU dinâmico e um algoritmo de previsão de RSSI, usados para melhorar a performance do Wi-Fi marítimo. Os resultados experimentais mostram ganhos consideráveis para regimes de baixa potência de sinal quando comparados com os obtidos apenas usando o MAC do standard IEEE 802.11.

Acknowledgments

"Why not Engineering?" Said my uncle once before I choose this master. This was the beginning of what would be my unique journey at FEUP.

I would like to thank Mário Lopes for his help, availability, and his specific way to make me answer to my own questions. Also, a special thanks to my Co-Supervisor Prof. Rui Campos for his support, motivation and commitment to every stage of this thesis. Finally to my Supervisor Prof. Manuel Ricardo for the opportunity of experiencing the best team and structure I have ever had in INESC PORTO.

I would also like to thank the BEST CASE project ("NORTE-07-0124-FEDER-000058" and "NORTE-07-0124-FEDER-000060") financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

For all the great moments, holidays and support, I would like to thank all my colleagues and friends, specially the closer ones which it is not necessary to name. Also, I want to leave a special thanks to my favorite lawyer Caetana for hearing me, even when she was not understanding a single word of what I was saying.

Finally, I would, obviously, like to mention my family, specially, my parents Irene and Joaquim for, beyond lots of other contributions, making it possible for me to accomplish my degree and for never allowing me to give up. Thank you.

João Magalhães

"In Nature nothing is created, nothing is destroyed,
everything is transformed"

Antoine Lavoisier

Contents

List of Figures

List of Tables

Abreviations and Symbols

ACK	Acknowledgment
AWGN	Additive White Gaussian Noise
A-MPDU	Aggregate MAC Protocol Data Unit
A-MSDU	Agreggate MAC Service Data Unit
BER	Bit Error Ratio
BS	Base Station
CDF	Cumulative Distribution Function
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear To Send
DCF	Distributed Coordination Function
DIFS	Distributed Inter-Frame Space
DSSS	Direct sequence spread spectrum
FEUP	Faculdade de Engenharia da Universidade do Porto
FHSS	Frequency-hopping spread spectrum
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HDD	Hard Disk Drive
HF	High Frequency
LOS	Line Of Sight
LTE	Long Term Evolution
MAC	Medium Access Control
mcs	Modulation and Coding Scheme
MIMO	Multiple In Multiple Out
MMCX	Micro-miniature coaxial
MPDU	MAC Protocol Data Unit
MS	Mobile Station
MSDU	MAC Service Data Unit
MTU	Maximum Transmission Unit
NAV	Network Allocation Vector
NTP	Network Time Protocol
NTPD	Network Time Protocol Daemon
ns	Network Simulator
OFDM	Orthogonal Frequency Division Multiplexing
OS	Operating system
OSI	Open Systems Interconnection
PLCP	Physical Layer Convergence Protocol
PLP	Packet Loss Probability
PoE	Power Over Ethernet

QoS	Quality of Service
RMS	Root Mean Square
RSSI	Received Signal Strength Indication
RTS	Request To Send
RTT	Round Trip Time
SIFS	Short Inter-Frame Space
SNR	Signal-Noise Ratio
TCP	Transmission Control Protocol
TID	Traffic Identifier
TVWS	Television White Spaces
UDP	User Datagram Protocol
VHF	Very High Frequency
WAAS	Wide Area Augmentation System
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Chapter 1

Introduction

1.1 Context

As the Internet is growing, new services are popping up. Some of them are becoming critical to every company, leading to the always-on connectivity requirement. In particular, low-cost broadband maritime communications are increasingly a need for fishing and other maritime companies. As stated in [10] and [12], available services such as satellite, HF/VHF and cellular networks (GSM, 3G, LTE) [8] are usually not affordable for most of them, are only available near shore, or are limited to voice communications.

Technologies such as WiMAX and Wi-Fi are emerging as a basis for broadband maritime communications, but problems have been identified in these environments. Using WiMAX at 5.8 GHz, presence of multipath, Doppler shift and boat's rocking causes delay and high Bit Error Ratio (BER) [6]. Experimental results using point-to-point, ship-to-ship communications, also showed that Wi-Fi at 5.8 GHz using IEEE 802.11n standard are possible within a 7 km range with, at least, 1 Mbit/s throughput despite the high measured delays and abrupt variations observed [12]. Using 700 MHz band [10], same minimum throughput was observed for distances up to 8 km from shore.

Sea oscillation along with reduced and unstable Received Signal Strength Indicator (RSSI) are two of the major issues identified; they lead to high Bit Error Ratio (BER) which causes high Frame Error Ratio.

1.2 Motivation

Small fishing companies and autonomous platforms cannot afford satellite services. Due to its isolation, their infrastructures are vulnerable and can be damaged or attacked. Real time video transmissions, VoIP communications and Internet access would bring efficiency, safety and comfort to those companies owners and employees. Even recreation crafts, such as yachts, could have Internet access and real time applications to increase comfort and connectivity.

Figure 1.1: Broadband maritime network

Also, sea-related research works, such as marine biologists works or sea monitoring, could use broadband low cost communication services, such as Wi-Fi, so that real time video transmissions, large scale data exchange and access to Internet servers could be possible. Thus, crucial information might be consulted anytime during sea field investigations and, in addition, research work could be immediately broadcast and kept safe in dedicated servers. For instance, already implemented, sea sensors can take environmental samples within certain time intervals, obtained data could be retrieved and consulted anytime if such service was available and affordable. A low cost, license free, broadband network supporting real time high bitrates such as a Wi-Fi based, would contribute to enable such services in the maritime environment.

WiMAX operating in the 5.8 GHz band has been tested in scenarios such as the presented in Figure 1.1, as it provides long ranges and high bitrates. Yet, WiMAX is a proprietary technology and typically requires big installation and operation costs due to high power characteristics and so, strong electrical support. Experimental tests using Wi-Fi have also been carried out. Wi-Fi operating at the 700 MHz band (TVWS) [1] was already tested to enable this kind of communications [10]. Also, in [12] it is shown that Wi-Fi 802.11n operating in the 5.8 GHz band can provide high bitrate real time transmissions. However, future work in these areas is still needed to adapt the Wi-Fi technology to the maritime environment and obtain better performance than the observed in [10] and [12].

1.3 Objectives

The main objective of this work is to suit the Wi-Fi based communications MAC for long distance and low signal strength communications. The idea is to create a dynamic MTU mechanism.

[1] Released television band due to digital television transition

This algorithm should work as a function of RSSI, a variable whose variations are hard to be monitored only with the wireless card information. Its development targets the maritime environment but may be extended to other communications having large BER. To achieve the main objective, the following intermediate goals are considered:

- Analyze experiments done using Wi-Fi 802.11n at 5.8 GHz and identify main problems encountered when dealing with broadband maritime communications;
- Build a new efficient RSSI prediction algorithm capable of monitoring this variable during time intervals where it is not read;
- Design a new algorithm for low RSSI, focused on dynamic MTU (packet length), bringing MAC improvements;
- Implement the two algorithms under Linux/OpenWrt;
- Test the new maritime communications solution in laboratory environment and in the INESC TEC's MARBED testbed;
- Analyze and compare obtained results with results obtained in previous tests using non improved MAC.

1.4 Contributions

The main contribution provided by this dissertation is a melioration for Wi-Fi 802.11n MAC, capable of improving communication characteristics, such as throughput, packet loss and delay. This contribution can be separated into two different strands:

- A RSSI prediction algorithm which receives discrete values as input and predicts the behavior of that variable during a given time;
- A dynamic MTU algorithm which changes the MTU according to the frame error probability estimated using the RSSI prediction algorithm.

1.5 Structure

This document is organized in 5 chapters. Chapter 2 describes the state of the art and presents the related work. Chapter 3 introduces the proposed solution. Chapter 4 presents the solution implementation, evaluation and experimental results. Finally, Chapter 5 draws the conclusions and points out the future work.

Chapter 2

State of The Art

In this chapter the solutions proposed so far to implement maritime broadband communications are presented, including solutions proposing modifications to the MAC in wireless sensor networks and WiMAX based solutions operating in the maritime environment. Also, packet length adaptation in Wi-Fi is addressed, since it brings up methods used in networks with poor radio links and represents an approach that can be used or adapted to the maritime environment. IEEE 802.11n standard is also addressed, since this dissertation will be based on this technology. Finally, the most used 2-ray maritime propagation model is presented in order to understand the sea behavior and its effects on maritime wireless links.

2.1 IEEE 802.11

IEEE 802.11 standards targets MAC and Physical layer. The physical layer started to be based on FHSS and DSSS; nowadays, OFDM is used. The MAC layer is based on listen-before-send techniques, it implements Career Sense Multiple Access with Collision Avoidance (CSMA/CA), which, rather than detecting collisions, such as in IEEE 802.3 wired networks, avoids them and waits for a backoff interval before each frame transmission [2].

DCF (Distributed coordination function) based on CSMA/CA shall be implemented in all stations, since it is the fundamental access method of the IEEE 802.11 MAC [11]. Before transmitting, a station shall sense the medium to determine if any other station is transmitting. If the medium is idle during a given interval (DIFS), the station can transmit after that interval elapses. If the medium is sensed busy, the station must defer the transmission so collisions can be avoided; after deferral and prior to attempting to transmit again, the station must select a random backoff time interval and wait that time until the medium is sensed idle again. Except for broadcast frames, where no ACK is required, a frame transmission is considered to be successful when an ACK is received by the sending STA from the destination station. To minimize collisions, a refinement of the method may be used based on a RTS/CTS handshake. After the medium is sensed idle and deferral and backoff time intervals are awaited, the sender transmits a RTS (Request To Send) and

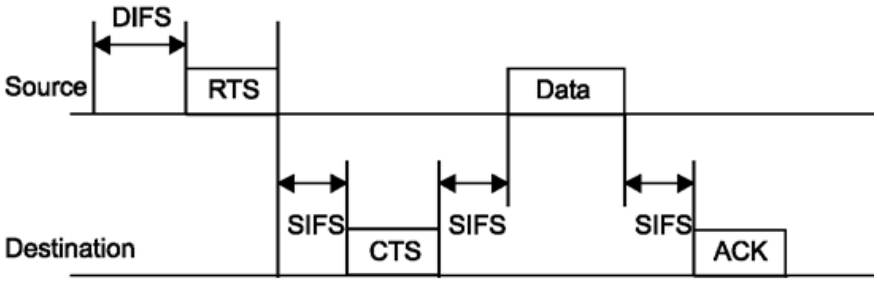

Figure 2.1: RTS/CTS/data/ACK and NAV setting [11]

waits for a CTS (Clear To Send), as illustrated in Figure 2.1. If the CTS is received, the sender is clear to transmit.

Due to the use of MIMO techniques, IEEE 802.11n offers data rate up to 600 Mbit/s, if four antennas are used at the sender and the same number is used at the receiver. IEEE 802.11n supports frame aggregation and reduced interframe spacing, which allows ACK blocking, thus, one single block ACK can be sent to indicate whether frames were transmitted successfully or not; also, idle periods between frames are reduced.

There are two methods proposed in IEEE 802.11n to perform frame aggregation: 1) aggregate MAC protocol service unit (A-MSDU); aggregate MAC protocol data unit (A-MPDU). A-MPDU differs from A-MSDU due to be processed after the MAC header encapsulation process. The two-level aggregation, comprising a blend of both, should be the most efficient. The basic operations of this comprising are illustrated in Figure 2.2 and can be summarized as follows:

- If there are several small MSDUs (payload) with same Traffic Identifier (TID), meaning they have the same priority, ready to be sent to the same receiver, a layer at the top of the MAC buffers them to form an A-MSDU (A-MSDU is complete either when the waiting packets reach maximal A-MSDU threshold or when the maximum preconfigured delay from the first buffered packet is reached);
- If TIDs of MSDUs are different, they move to the second stage where they are packed together with any A-MSDUs derived from the first stage or other single MSDUs (then being considered MPDUs) by using A-MPDU aggregation (A-MSDU restriction of aggregating frames with matching TIDs is not a factor with A-MPDUs due to be processed after the MAC header encapsulation process).

It must be mentioned that, using two-level aggregation, the maximum MPDU supported by an A-MPDU is 4095 bytes. Then, A-MSDUs or MSDUs with lengths larger than this cannot be transmitted. A-MSDU and A-MPDU aggregation mechanisms can be used separately, even though that is less efficient and introduces processing delays and complex aggregation. For instance, while using A-MSDU, if any MSDU is corrupted, the entire A-MSDU must be retransmitted which may lead to higher delays than the ones obtained without aggregation.

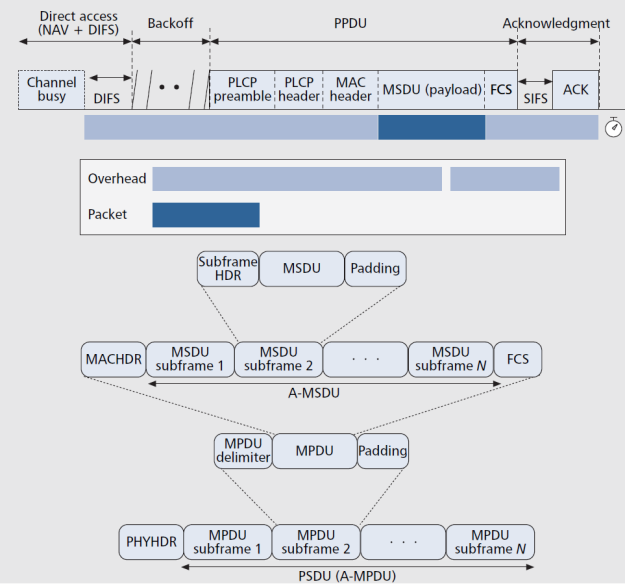

Figure 2.2: DCF basic operation vs Two-level frame aggregation [4]

2.2 Maritime Radio Propagation Models

Due to constant oscillation, reflections and refractions, the maritime environment presents unique radio propagation conditions. In [5] several measurements using WiMAX at 2.5 GHz in a seaport scenario were performed to evaluate their influence on communications. The authors stated that a 2-ray pathloss model, based on the direct ray and the reflected one, is able to fit the actual behavior of the observed maritime wireless channel. Simulation studies, carried out using OPNET simulator [1], have shown that the 2-ray pathloss model could reproduce measured values, despite not being able to predict all environment behaviors.

Figure 2.3: Site of installation [5]

[1] OPNET Technologies, Inc. is a software business that provides performance analysis for computer networks and applications

The testbed (Figure 2.3) consisted of a land sectoral antenna, placed 30 meters above the sea surface, with a gain of 17 dBi transmitting at 35 dBm of power and a 11 dBi omnidirectional antenna set at approximately 10 meters height placed at the ship. The system performance was evaluated in terms of RSSI and throughput by means of UDP sessions. The results showed some deepholes up to 5 km distance (Figure 2.4), due to characteristics of maritime environment, which differentiates it from free space model. After 5 km, RSSI looked quite stable with a linear decrease of less than 1 dB/km, after 19 km far from the shore, the communications broke down.

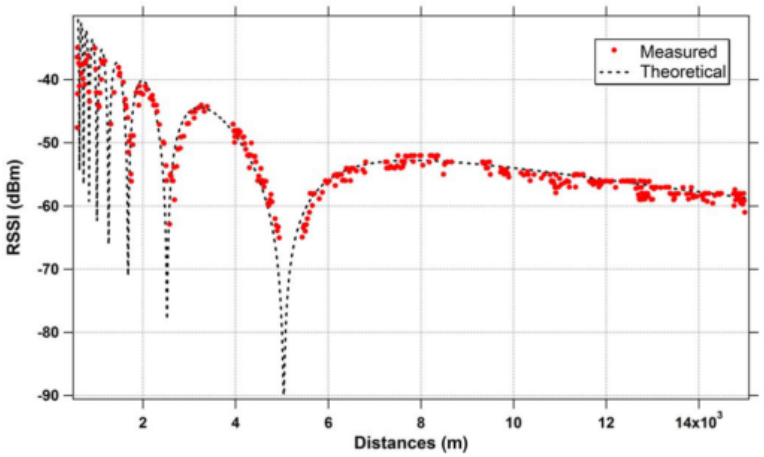

Figure 2.4: Theoretical and measured RSSI for different distances [5]

According to measured RSSI, the following 2-ray model was developed:

Power Received:

$$P_r = \frac{P_t G_t G_r}{L_2 ray} \tag{2.1}$$

Proposed 2-ray Pathloss Model:

$$L_2 ray = \frac{L_f s}{\beta} \tag{2.2}$$

Free Space Pathloss Model:

$$L_f s = \frac{4\pi d^2}{\lambda} \tag{2.3}$$

Reflection Coefficient:

$$\Gamma(\theta_i, n_1, n_2) = \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} \tag{2.4}$$

$$\theta_t = \arcsin(\frac{n_1}{n_2}\sin\theta_i) \tag{2.5}$$

$$\beta = 1+\Gamma(\theta_i,n_1,n_2)^2 - 2\Gamma(\theta_i,n_1,n_2)cos(\frac{4\pi h_t h_r}{\lambda d}) \tag{2.6}$$

- θ_i represents wave angle of incidence, $n_1 = 1$ represents refraction index of air and $n_2 = 1.333$ represents refraction index of water. P_t, G_t and G_r represents transmission power, transmitter antenna gain and receiver antenna gain respectively. d and λ represent distance, between base station BS and mobile stations, MS and wavelength respectively.

Comparing 2-ray pathloss model with the free space pathloss model, several similarities can be found, except on the 2-ray pathloss peaks. This is due to specific sea characteristics which no other model available in OPNET was able to capture, as shown in Figure 2.5.

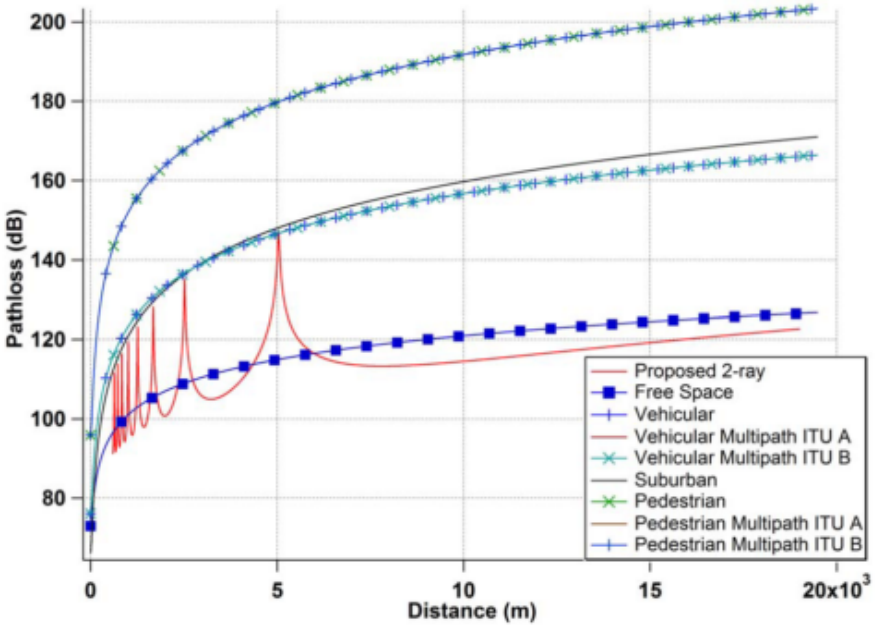

Figure 2.5: Simulated pathloss model results [5]

2.3 Wi-Fi Maritime Communications

Wi-Fi is taking its first steps in long range maritime communications. In this section we refer to the results obtained using Wi-Fi in the 700 MHz (TVWS) and 5.8 GHz bands. In both cases, it was proved that it is possible to communicate on maritime environment using this technology. Results revealed stable throughput within a 9 km range, using TVWS band; using 5.8 GHz band, at least, throughput of 1 Mbit/s was observed up to 7 km, being registered more frequent connection losses after that distance. In order to perform these tests, one base station and one fishing ship were used. as illustrated in Figure 2.6. This testbed, named MARBED, is detailed in Section 2.3.1 and the conclusions of these two experiments are presented in Sections 2.3.2 and 2.3.3.

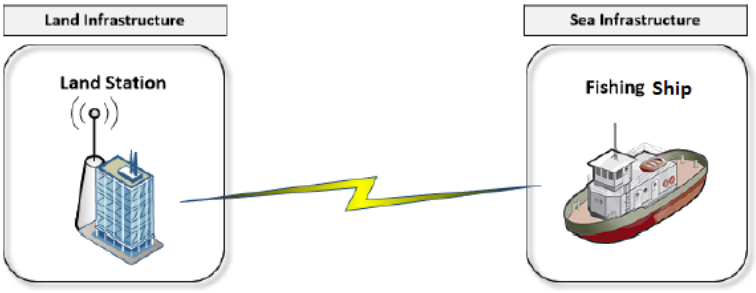

Figure 2.6: Overview of the entire Wi-Fi maritime networks testbed [10]

2.3.1 MARBED Testbed

For the time being, MARBED consists of a land station and a sea node installed in a fishing ship. Each communication node includes a single board PC with a Wi-Fi network interface card. The sea node also incorporates a GPS receiver in order to track the ship's position. An omndirectional antenna is used in the sea node and a sectoral antenna is installed in the land station (Figure 2.7).

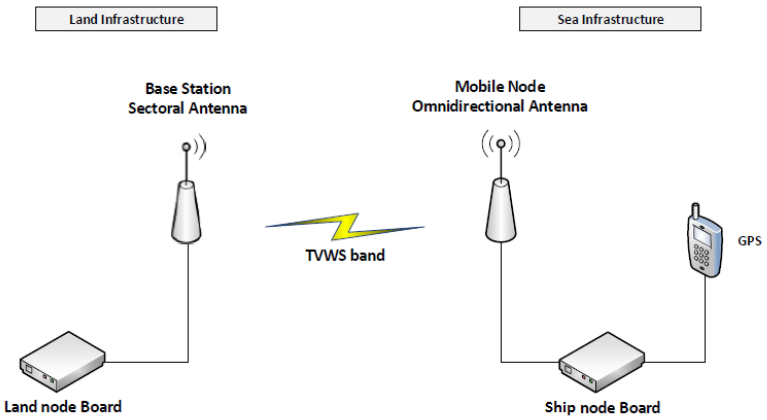

Figure 2.7: Specifications of the entire system [10]

In what follows, we detail the hardware and software specifications

2.3.1.1 Hardware Specifications

Both nodes consist of an Alix3d3 board (Figure 2.8), manufactured by PC Engines [1] containing two miniPCI slots, that can be for wireless cards. Low power consumption Alix3d3 (powered up via PoE), having a 500 MHz AMD Geode processor (x86 architecture) and 256 MB DDR RAM is suited for wireless routing and network security applications due to its robustness and the ability to install several Operating Systems and applications. The absence of Hard Disk Drive (HDD) avoids physical problems caused by environment conditions, namely boat's rocking and vibration, thus, used storage is a USB pen drive.

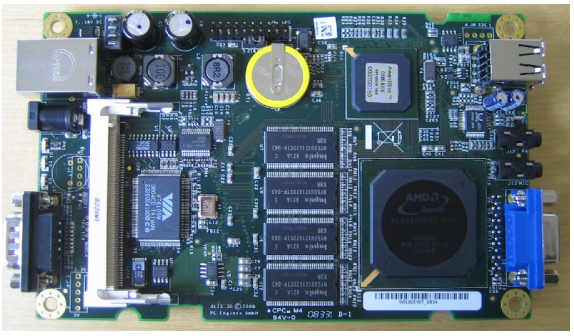

Figure 2.8: PC Engines Alix3d3

Both the land station and the sea node include a Low Power Consumption MikroTik RouterBOARD R52n-M [13] (Figure 2.9). This network adapter supports IEEE 802.11a/b/g/n at 2.4 GHz and 5 GHz bands, supporting throughputs up to 200 Mbit/s.

Figure 2.9: MikroTik RouterBOARD R52n-M

The GPS receiver installed in the sea node is showed in Figure 2.10. It contains a USB interface that enables data communications with the Alix board and avoids any external power source.

Figure 2.10: USGlobalsat [14] BU-353 USB GPS Receiver used in the sea node

In the land station, a sectoral antenna with 16 dBi gain and 120 degrees horizontal width is installed; the chosen antenna is model AM-5G16-120, manufactured by Ubiquiti [2]. In the sea node, an omnidirectional antenna with 10 dBi gain is installed.

Due to harsh climacteric conditions experienced in the maritime environment, a waterproof box along with cable protections were used as illustrated in Figure 2.11.

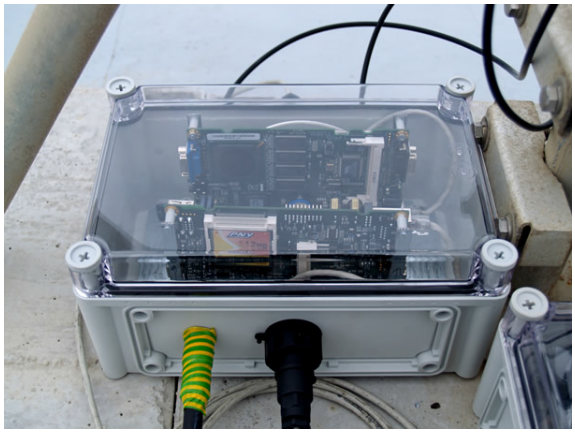

Figure 2.11: Waterproof box used to protect the Alix Board [10]

2.3.1.2 Software Specifications

The software considered in the maritime testbed is the following:

- **OpenWrt** [3] - Based on Linux Kernel, this operating system is primarily used on low processing, limited memory and storage systems to route network traffic. It is supported by

[2] http://www.ubnt.com
[3] https://openwrt.org/

chosen hardware and meets key requirements of this work, namely, full customization of the wireless card. In addition, this operative system offers a web interface in order to facilitate its configuration.

- **Wireless-Tools** - This is a packet available on OpenWrt containing several network applications. In this work, the most important is IWinfo which allows to monitor the wireless link.

- **Iperf** [4] - Through traffic generation, this application offers mechanisms which enables evaluation of TCP/UDP bandwidth performance. Allows the tuning of several parameters and UDP characteristics, reporting factors such as bandwidth, delay, jitter and datagram loss.

- **GPSD and CGPS**[5] - Commonly used on Linux systems, GPSD is a daemon which translates and provides data obtained from GPS receiver through a defined port. CGPS, monitors data in that port giving the user a friendly interface with positioning details.

2.3.2 Experimental Tests Using TV White Spaces band

TV White Spaces Band (700 MHz) is recently available due do migration to the digital television service. This band is characterized by longer transmission range and better propagation. These characteristics make this band suitable for long range communications [10], namely maritime ones. Wi-Fi over TV White Spaces is now a standard with the ratification of IEEE 802.11af standard in February 2014.

> "This dissertation presented the first measurements done using the IEEE 802.11b/g Wi-Fi and TVWS band in maritime environment." [10]

Using this system, Wi-Fi maritime communications tests were performed. Considering as performance metrics, throughput, round trip time (RTT), packet loss ratio, received signal strength indicator (RSSI), and jitter, the following conclusions were drawn [10]:

- Although it was possible to measure RSSI about 17 km from the shore, the maximum distance where a stable link was maintained was 13 km, although data was transfered at about 15 km from shore with residual TCP throughput of 18.5 Kbit/s;

- It was also observed that up to 8 km, throughput was about 1 Mbit/s, sharply decreasing after this distance and leading to a maximum range, having reasonable link conditions, of 9 km;

- UDP showed better throughput, range and tolerance against signal obstruction when compared to TCP.

[4] http://iperf.fr/
[5] http://gpsd.berlios.de/

- Jitter measurements revealed a gradual increase up to 150 ms after 10 km, which are critical values for real-time applications.
- Application layer packet loss ratio was almost 0% up to 9 km.

It was concluded that the propagation model in the maritime environment followed a two ray path loss exponent and that, within 9 km, Wi-Fi communications links are possible high stable throughputs. Boat's rocking incurring RSSI abrupt variations, multipath and Fresnel zone obstruction were identified in this work as critical problems in maritime communications.

2.3.3 Experimental tests Using 5.8 GHz band

Standards IEEE 802.11a and IEEE 802.11n may operate in the 5.8 GHz band. Due to physical layer codification and the use of MIMO, higher throughput is achieved so, IEEE 802.11n was chosen in [12] to study Wi-Fi maritime communications at 5.8 GHz.

Two MAC optimizations were applied in these experimental tests, acknowledgment (ACK) disabling and frame aggregation. The first one may lead to higher throughputs by reducing waiting time for ACK to zero, although packet loss may occur and it is not reversible. However, in long range communications, this optimization may be used. Frame aggregation enables the use of one header for several packets allowing greater throughput [12].

During experimental tests under calm sea and reasonable atmospheric conditions, the performance results obtained in the tests were:

- Up to 7 km range, communication is possible with satisfactory quality of service (throughput, jitter);
- When too close to shore, boat's rocking and small vertical aperture of the antenna used in the sea node made the link break down frequently;
- Using TCP, packet confirmation was necessary. In these conditions, lower throughput was observed when boat is around 7 km far away from the coast;
- For UDP traffic, throughput as a function of distance results are illustrated in the plot from the Figure 2.12;
- Round-Trip-Time (RTT) analyses using Ping showed that only 6 packets from 10 were delivered at 6400 meters, this is either due to packet loss during the request or due to packet loss associated to reply (response from a station B to a request from a station A).

These tests showed that, after 6.5 km away from shore, sea oscillation and low signal strength links were responsible for connection losses. On the other hand, close to what was observed in Section 2.3.2, 1 Mbit/s Wi-Fi on 5.8 GHz band transmissions were possible up to 7 km from shore [12].

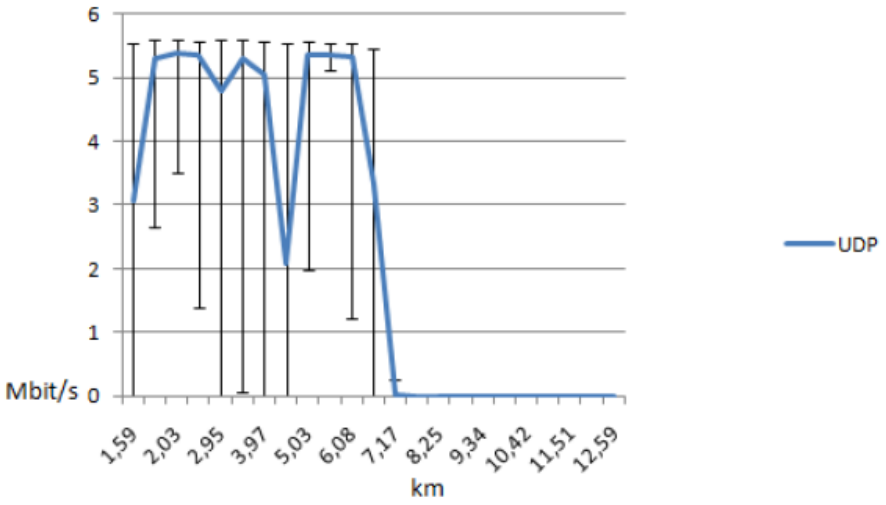

Figure 2.12: UDP throughput vs distance [12]

2.4 WiMAX Channel exploitation for maritime communications

> "Broadband wireless access for maritime users is a brand-new topic which demands sophisticated transmission technology to support high speed and wide coverage communications." [16]

In [16], the authors focused on IEEE 802.16 (WiMAX), maritime communications, due to its high transmission rate, wide coverage, mobility support and flexible deployment, and developed a new transmission scheme capable of exploit maximum connection capacity and bypass sea movements and respective connection losses. The authors state that there are 3 attenuation factors present in maritime communications channel: path loss, path shadowing and multipath fading. The first one is due to long distance transmissions and connection failures, although the link is established in LOS (Line of Sight). Path shadowing is caused by the presence of large obstacles between two communicating nodes. Multipath fading is due to reflections received together with direct link. According to the typical sine wave, which can be used to model the movement patterns of the sea surface, a node in the sea goes up and down periodically, as illustrated in Figure 2.13. $X_i(t)$ and $Y_i(t)$ denote the horizontal and vertical coordinates of node i and a; λ and T denote wave amplitude, wave length and wave cycle of sea's surface respectively. The equation representing vertical position of node B is:

$$Y_B(t) = a\sin[2\pi * \frac{mod(X_B(t),\lambda)}{\lambda} + 2\pi * \frac{mod(t,T)}{T}] \tag{2.7}$$

Along with this equation, in [16], an equation used to estimate horizontal coordinate of wave peak in terms of time is presented, allowing to predict the moment when boat's position matches

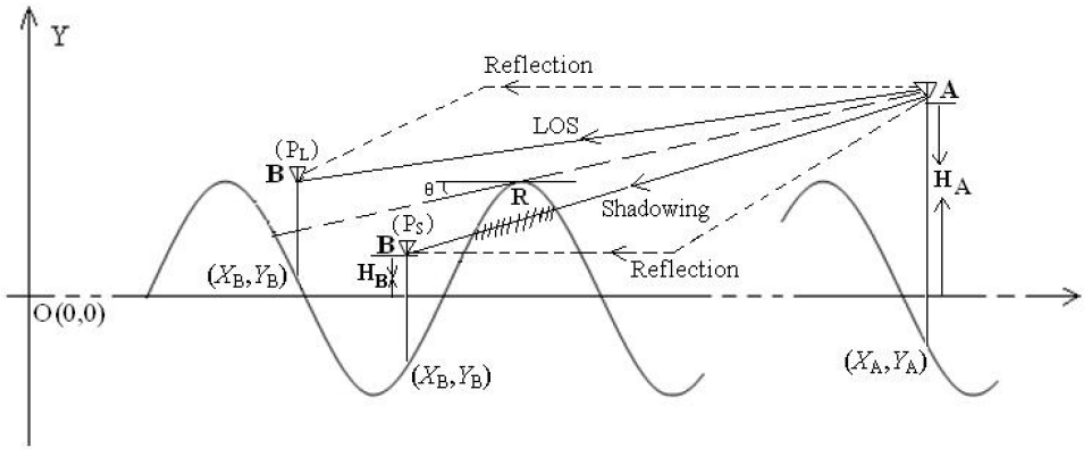

Figure 2.13: Maritime channel modeling [16]

the wave peak. In addition, Doppler Effect and SNR are also estimated and taken into account in order to predict transmission rates within a certain time slot.

Simulation results were based on 2 different algorithms used to exploit available predicted bandwidth in a scenario composed by several mobile sea nodes. One of them is ATS (Adaptive Time Scheduling) which tries to serve all nodes and the other is GTS (Greedy Traffic Scheduling) which serves stations with better links making the others "starving". In addition, scenarios with moving and stuck boats and two different maritime surface conditions were taken into account. By calculation link parameters within a certain time period, simulation results showed performance and stability (e.g., throughput) improvements in a maritime wireless network which are reflected in better QoS metrics.

2.5 SEA-MAC: Simple energy aware sea MAC protocol

In [3], the authors developed an energy saver MAC for wireless sensor networks (SEA-MAC) used in maritime environment monitoring applications. These networks are composed by battery-operated nodes; so, power saving is critical [3]. In these experiments, nodes wake up only when sensors take environment samples and sensors synchronization is controlled by the base station. By avoiding idle listening between two samples, this mechanism allows energy saving and avoids sleep/listen mechanisms implemented in other sensor protocols. Thus, it is suitable not only for maritime sensor networks but also for other sensor networks.

SEA-MAC was meant to compete with other MAC protocols developed for sensor networks, S-MAC and SCP-MAC, considering sea unique proprieties and the need to lower power consumption. SEA-MAC was simulated over ns-2 [6] and implemented in mica2 motes running TinyOS

[6]Network Simulator, version 2

[7]. The protocol was evaluated against other MACs in terms of energy consumption and delay. Simulation results showed that SEA-MAC achieves better energy performance and slightly better delays than its counterparts.

2.6 Packet length adaption in WLANs

In [9] the authors state that it is possible to avoid packet loss by reducing packet length at the MAC layer. Based on ns-2 simulations, it is also concluded that 20% throughput gains are possible. In the proposed approach, packet length is adapted by each node based on shared informations about channel occupancy by the access point so that collision probabilities can be estimated.

Variable packet length is supported by IEEE 802.11 standards and so, setting the maximum packet length to higher values can reduce the impact of the overhead. This is more suitable when there is a single strong link between two nodes. In a scenario with weaker link and hidden nodes, shorter packets may be a preferable solution. Bigger packets lead to more symbols decoding, thus they are more susceptible to packet loss due to more frequent channel errors. Dynamic packet length seeks to address the tradeoff between lower overheads for bigger packets and lower packet loss for smaller packets (Figure 2.14).

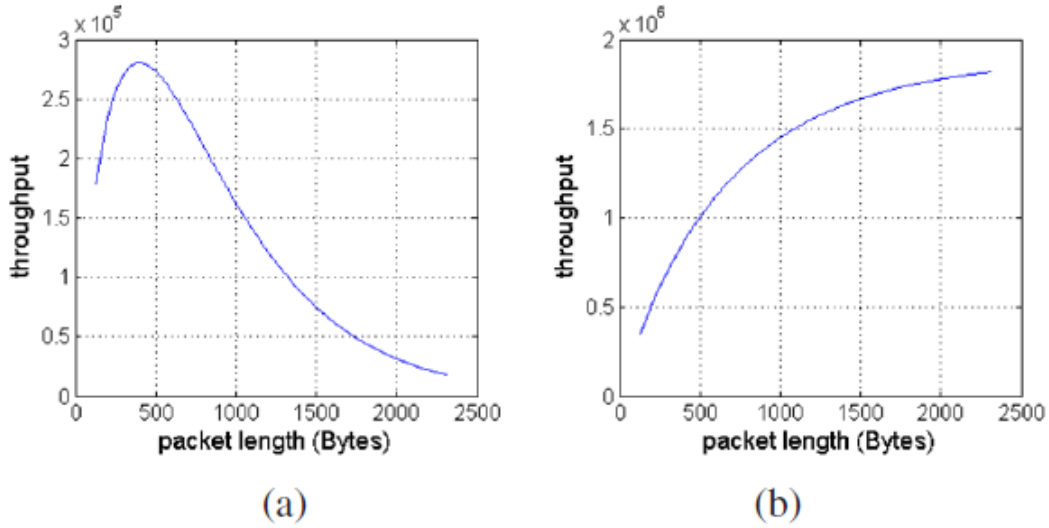

Figure 2.14: Throughput vs packet length assuming no collisions for (a) SNR fixed at 9 dB, and (b) SNR with mean of 9 dB with standard deviation of 3 dB. [9]

Losses in Wi-Fi networks can be due to collisions or transmission errors. Collisions may occur due to traffic overlap when two nodes are transmitting simultaneously. Transmission errors may derive from low SNR, which is either due to large distances between two nodes or due to multipath fading. The total packet loss probability can be computed as:

[7] Wireless Smart Sensors http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf

$$P_L = 1-(1-P_C)(1-P_e) \tag{2.8}$$

where P_C is the collision probability and

$$P_e = 1-(1-BER_{Rh}(\sigma))^{L_h}(1-BER_{Rp}(\sigma))^{L} \tag{2.9}$$

where L and L_h are the lengths of the header and the payload respectively, R_h and R_p represent the modulation rates for header and payload and $BER_{Rx}(\sigma)$ is the Bit Error Ratio relative associated to each one of them and to Signal Noise Ratio (SNR) σ.

The simulation tests in [9] showed packet length adaption mechanisms were not able to detect rapid channel changes, although it was proved intelligent packet length selection based on the estimation of channel distribution could yield gains with respect to using maximum, static packet length. It is relevant to note that this paper targeted hidden nodes and time-varying channels problems.

2.7 Bit Error Ratio

From formula 2.9 remains undercovered the estimation of BER. In [7] the authors present a study and implementation of IEEE 802.11 Physical Channel Model in ns3 (Network Simulator 3), thus, BER per modulation is one of the covered subjects. Referring AWGN channels, BER for each modulation is given by:

- BPSK

$$BER = Q\left(\sqrt{2\frac{E_b}{N_0}}\right) \tag{2.10}$$

- QPSK

$$BER = \frac{1}{2}(2Q\left(\sqrt{2\frac{E_b}{N_0}}\right)-Q^2\left(\sqrt{\frac{E_b}{N_0}}\right)) \tag{2.11}$$

- M-QAM

$$BER = \frac{1-\left(1-\frac{2(\sqrt{M}-1)}{\sqrt{M}}*Q\left(\sqrt{3log2(M)\frac{\frac{E_b}{N_0}}{M-1}}\right)\right)^2}{log2(M)} \tag{2.12}$$

where

$$\frac{E_b}{N_0} = SNR \frac{signal\_spread}{modulation\_rate} \tag{2.13}$$

and $Q(x)$ represents $Qfunc(x)$.

2.8 Prediction Models

This section addresses two prediction algorithms available in the state of the art: Kalman model and Grey Model. These models were chosen since it was assumed they are suited for RSSI prediction since all the consulted references related to this matter address them. The significance of this topic will be clear upon the presentation of our proposed solution in Chapter 3.

2.8.1 Grey Model

In the System Theory there is the differentiation between *white system* and *black system*. They are characterized for knowing all system information or the lack of it, respectively. *Grey System*, also known as Grey model, is situated between those, it "mainly works on a system analysis which has produced poor, incomplete or uncertain messages" [15]. Grey Model requires a few data for system estimation and it is easily mathematically modeled. GM(1,1) is the most commonly used Grey Model. One variable system's behavior can be modeled by a first-order differential equation. For example, a set of RSSI values can be used as Grey's input, being its output *RSSI* predicted values. Modeling procedure is as follows:

Given the original data set:

$$RSSI[1], RSSI[1], ...RSSI[n] \tag{2.14}$$

Where RSSI[x] represents measured signal strength as system input at time x, a new sequence

$$RSSI_1[1], RSSI_1[1], ...RSSI_1[n] \tag{2.15}$$

where

$$RSSI_1[k] = \sum_{m=1}^{k} RSSI[m] \tag{2.16}$$

is estimated.

From $RSSI_1$, first-order differential equation can be formed

$$\frac{dRSSI_1[k]}{dk} + aRSSI_1[k] \quad = \quad u \tag{2.17}$$

Being possible to obtain

$$\begin{bmatrix} a \\ u \end{bmatrix} \quad = \quad (B^T B)^{-1} B^T RSSI'_n \tag{2.18}$$

where

$$B \quad = \quad \begin{bmatrix} -\frac{1}{2}(RSSI_1(1) + RSSI_1(2)) \\ -\frac{1}{2}(RSSI_1(2) + RSSI_1(3)) \\ \dots \\ -\frac{1}{2}(RSSI_1(n-1) + RSSI_1(n)) \end{bmatrix} \tag{2.19}$$

and

$$RSSI'_n \quad = \quad \begin{bmatrix} RSSI(2) & RSSI(3) & \dots & RSSI(n) \end{bmatrix}^T \tag{2.20}$$

Finally, the prediction function is $RSSI_{1a}(k) = (RSSI_1(1) - \frac{u}{a})e^{-ak} + \frac{u}{a}$ which for instant $k+1$ can be written as

$$RSSI_a(k+1) \quad = \quad (RSSI(1) - \frac{u}{a})(e^{-a} - 1)e^{-ak} \tag{2.21}$$

corresponding to the predicted value, given the input values of Equation 2.14.

2.8.2 Kalman Model

For Kalman Modeling, a Kalman Filter is considered where a set of RSSI values are known as well as some information of it. Kalman Filter is very powerful, supporting past, present and future estimation. Although the computation in C language of this algorithm were performed and tested, its procedures will not be addressed in this dissertation since it was understood that there was not enough information about the system to introduce as filter input. For this reason, results from its comparison with Grey model were not satisfactory bringing up more accurate prediction from the second model. This justifies Kalman filter prior exclusion and so, its non implementation and use in the proposed solution.

2.9 Summary

IEEE 802.11n standard supports MIMO which can be used to achieve high throughputs using multiple antennas. In our case, MIMO may not be contemplated, although other characteristics from IEEE 802.11n such as frame aggregation and ACK blocking are suitable.

Two pioneer experiments based on land-sea Wi-Fi communications were presented, along with INESC TEC's maritime testbed (MARBED). Environmental derived issues such as boat's rocking, multipath, Fresnel zone effect and Doppler shift were identified. Both Wi-Fi researches (5.8 GHz and 700 MHz) using MARBED showed the presence of gaps regarding Wi-Fi maritime communications, creating possibilities for future research work

With the maritime 2-ray pathloss model presented in Section 2.2, it was clear that sea environment introduces new variables in the free space model.

Also, it was understood that MAC improvements should be capable of minimizing Packet Loss Probability occurring in harsh maritime conditions and poor link state, thus the reduction of packet size based on RSSI was seen as a possible bottom line. Research for Packet Loss Probability lead to the formulas presented in Section 2.6 used to estimate it. On the other hand, in order to estimate BER parameter presented in those formulas, it was necessary to describe the models in Section 2.7.

Finally, the need of monitoring RSSI lead us to the prediction models presented in Section 2.8 necessary due to the requirement of having RSSI information available between RSSI readings.

Chapter 3

Proposed Solution

This chapter presents the proposed solution, which changes 802.11 MTU in order to keep packet loss probability lower than a defined threshold. Packet loss probability is estimated based on the Received Signal Strength Indicator (RSSI), which, besides being obtained through an OS variable (every one second), is predicted using a new algorithm proposed in this work as well. The prediction occurs during the one second interval when no RSSI value is being read from the network interface card. The prediction is used to know in advance the RSSI changes and, in that way, modify the 802.11 MTU at the moment it is, theoretically, necessary. This chapter addresses the prediction algorithm explanation along with the description of the dynamic MTU algorithm.

3.1 RSSI Prediction Algorithm

The prediction algorithm presented herein was defined using a trial and error iterative approach. Several versions of the algorithm were built and tested; when improvements were achieved, the modifications were adopted. The mechanism used to understand whether a version is better is explained in Section 4.1.

Analysis of logs from previous maritime Wi-Fi investigations (please see Section 2.3) lead us to model the RSSI variation as a step wave, as shown in Figure 3.1.

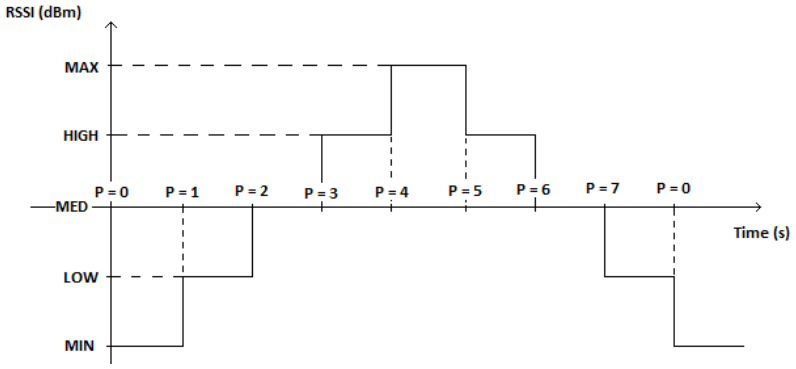

Figure 3.1: Step Wave designed for sinusoidal wave modeling

In the design of our algorithm, it is considered this wave is divided into five steps, Maximum, High, Medium, Low and Minimum. Being RSSI represented by negative values, in this wave absolute values are used, thus, Maximum step represents minimum RSSI value. For instance, the plot of Figure 3.1, possible values for steps from MIN to MAX could be: -76, -78, -80, -82 and -84. The 5 steps division was an assumption resulting from graphics observations which lead to the presented 8 parts division.

Given the step wave of Fig 3.1, as soon as an RSSI value is read from OS, our algorithm aims at discovering the position it represents in the step wave (positions vary from P = 0 to P = 7). Also, it is necessary to estimate maximum (MAX) and minimum (MIN) values as well as the period from the assumed step wave. The estimation of these parameters represents the first part of the algorithm. It was found that the best way to make it is represented by the flowchart Figure 3.2.

Flowchart begins with the reading and storage of a real RSSI values. It will stop reading when vector is fulfilled. Vector size (V_{MAX}) is four, this value was proved to be the best using a trial and error approach. This approach was based on a "for" cycle increasing V_MAX which, in each iteration, ran the algorithm for the same RSSI sample. V_MAX that lead to the closest results in relation to real values was four.

Each new RSSI read value is kept in the vector's last position. RSSI value is processed from wireless link information obtained from the wireless network interface by using, for example, the *Iwinfo* command:

iwinfo wlan0 info

The algorithm proceeds looking for MAX and MIN and estimating MED, HIGH and LOW by the following:

$$MED = \frac{MAX + MIN}{2} \tag{3.1}$$

$$HIGH = \frac{MED + MAX}{2} \tag{3.2}$$

$$LOW = \frac{MED + MIN}{2} \tag{3.3}$$

Now, if last value equals the previous, period is one second, being the positioning determined by the variation from the penultimate and ante penultimate values.

On the other hand, if last values are different, period is estimated by

$$period = \frac{| v[V_{MAX}] - v[V_{MAX} - 1] | + | v[V_{MAX} - 1] - v[V_{MAX} - 2] |}{2} + 1 \tag{3.4}$$

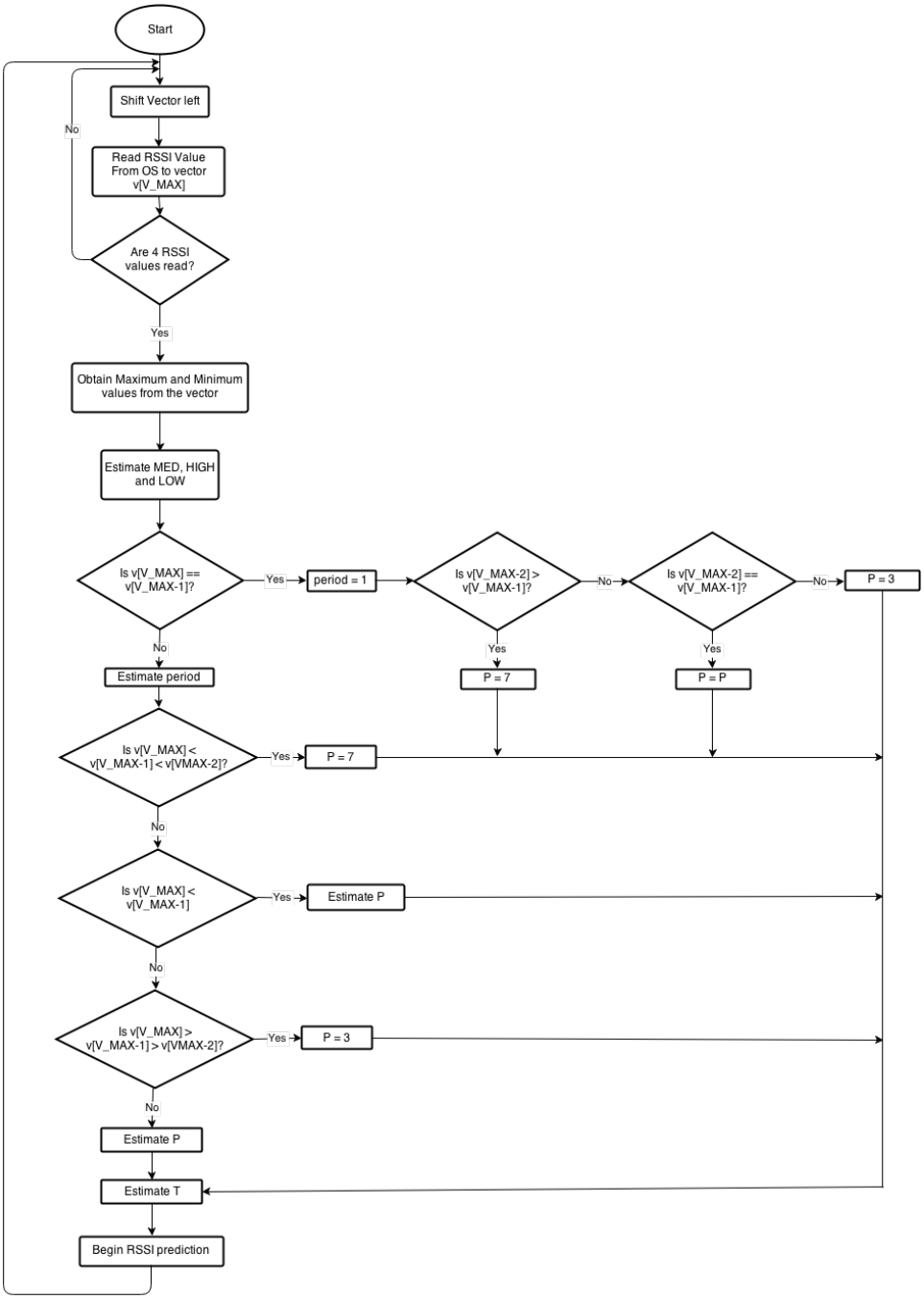

Figure 3.2: Wave conditions algorithm

This formula was improved consecutively, and represents the best of three developed ways to discover the period based on sea measurements observations. Firstly, the period was estimated subtracting the instants where the maximum and the minimum value were located in the vector. This was considered to figure half the time of the wave duration, being the period twice of it (if maximum was found on second one and maximum on second three, period was equal to four). Second approach counted the number of RSSI values above and below the mean of the entire

vector (this technique was developed for bigger V_MAX, about eleven). Period was considered to be the duration of the vector (for V_MAX equal to eleven, twelve seconds) divided by the mean of both those counts. Finally, used period estimation method considers that it increases proportionally with the increase of the difference between the value of two consecutive RSSI readings. Trial and errors methods also shown that results would improve if it was used the mean of the differences between last points and the previous. For instance, if the last three RSSI readings reveal values -81, -80 and -82 dBm, the difference is 1 and 2 respectively. The mean is 1.5 and the period would be 2.5. Since it is an integer variable, final period value would be 3 seconds.

P value is again based on the variation from the penultimate and ante penultimate values and its estimation is done by the formulas

$$P = 4 * \left(1 - \left(\frac{\mid v[V_{MAX}] - v[V_{MAX} - 1] \mid}{\mid MAX - MIN \mid}\right)\right) - 1 \tag{3.5}$$

if the last value is lower than the previous and

$$P = 4 * \left(\frac{\mid v[V_{MAX}] - v[V_{MAX} - 1] \mid}{\mid MAX - MIN \mid}\right) - 1 \tag{3.6}$$

if the last value is higher than the previous.

These two last formulas were developed in order to find out the position of the measured RSSI value, in the step wave. They were also designed after observing real RSSI measures and performing improvements and experiments with previous versions. The idea is to associate P with the relative difference between the two last measured points and the step wave amplitude. For example, if the maximum and the minimum values in the vector are -82 and -80 dBm, respectively, the difference between both is 2. Then, if RSSI last measures were the same as last example (-81, -80 and -82 dBm), since last value is higher (in module) than the previous, and lower than the first, P equals $\frac{2}{2} * 4 - 1 = 3$ representing one position before the top of the step wave.

Remains only T estimation (duration of each interval P) which is given by

$$T = \frac{period}{8} \tag{3.7}$$

Represented in the flowchart 3.2 by "Begin RSSI prediction", this task represents RSSI prediction mechanisms and is illustrated in the flowchart 3.3.

This part is responsible for predicting the received power during 1 second. Knowing all wave parameters above described, this part is focused on increment P and understanding the position it represents in the step wave, and so, the associated power (Figure 3.1).

This process begins setting time_remaining variable to one second. If T is bigger than one, it means that no prediction can be made because estimated P will remain in that step for more than one second. Then, time_remaining is set to zero and change_MTU is called with T=1. This

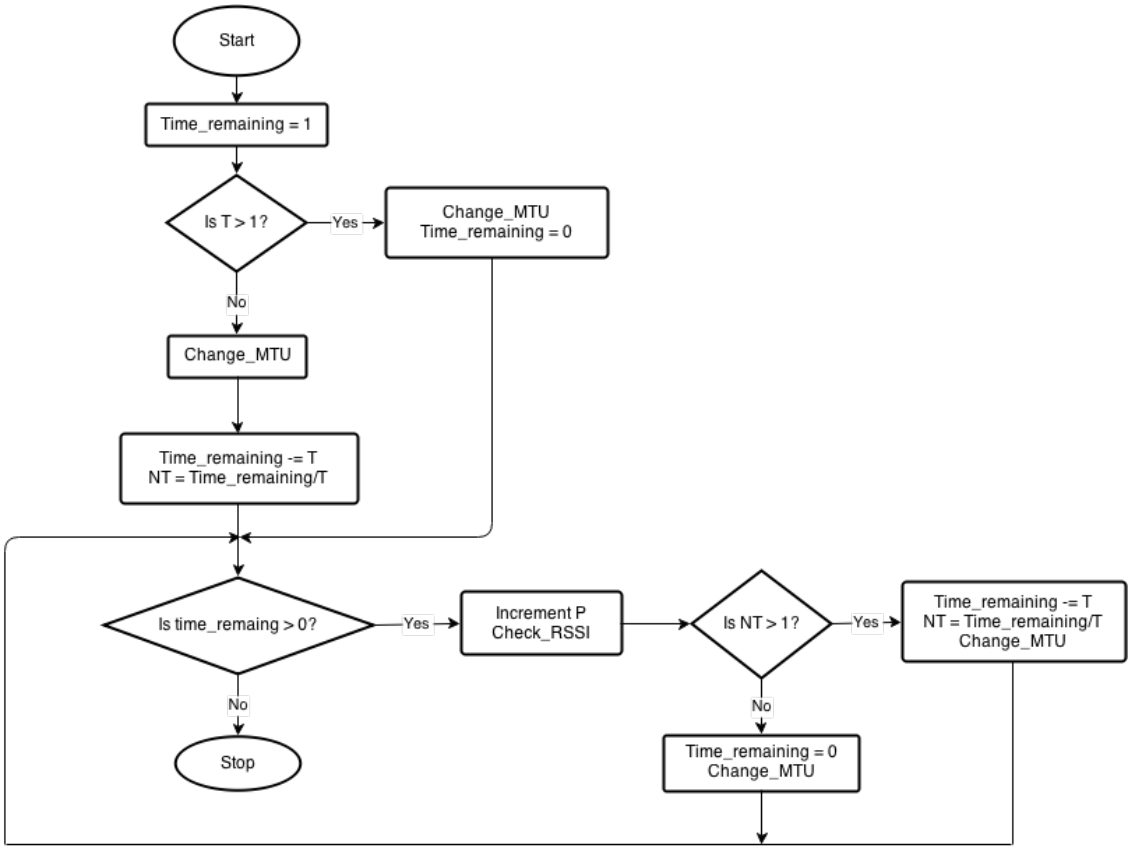

Figure 3.3: RSSI Prediction Algorithm

function represents the last part of the algorithm, presented in detail in Section 3.2, and requires parameters T and RSSI read from the network interface card. Thus, this function will know current signal strength and its duration in order to change MTU, if necessary. Even if T is not bigger than one, change_MTU is called. NT is estimated and represents the number of possible predictions in the remaining time. Time_remaining is consequently decremented by T until reaches zero.

In order to predict a new RSSI value, P is incremented, showing what will be next step wave position. Then, check_RSSI function is called with parameters P, MAX and MIN, returning the RSSI predicted value.

Now, if this there are no more time for previsions (NT was one), T, which is passed to function change_MTU, is $NT * T$, so the MTU is changed (if necessary) only during the remaining time until one second.

On the other hand, if more prediction iterations are available according to time remaining, change_MTU is called with T and the RSSI from check_RSSI function. Being then estimated new time_remaining and NT.

RSSI prediction algorithm runs for one second and finishes. A new RSSI value is read and the prediction algorithm starts a new estimation.

Using the example initiated on flowchart 3.2, we have P=3, period=3 and read value -82 dBm. T, in his case equals 0,375 seconds, change_MTU would be called with these values. Time_remaining would decrement by T to 0,625 and NT would be more than one. Now, for first value prediction, P is incremented to 4 which represents the last step of the step wave. Check_rssi will then return -82 which is the maximum (MAX) from the vector (last step). Next procedures are similar, P is again incremented and the value associated to its step is obtained, being used to possibly change MTU.

3.2 Dynamic MTU Algorithm

The change_MTU function referred in the flowchart of Figure 3.3 implements the dynamic MTU algorithm. By receiving RSSI value and the time interval T of the step, it decides whether to change or not the MTU. Flowchart in the Figure 3.4 presents this mechanism.

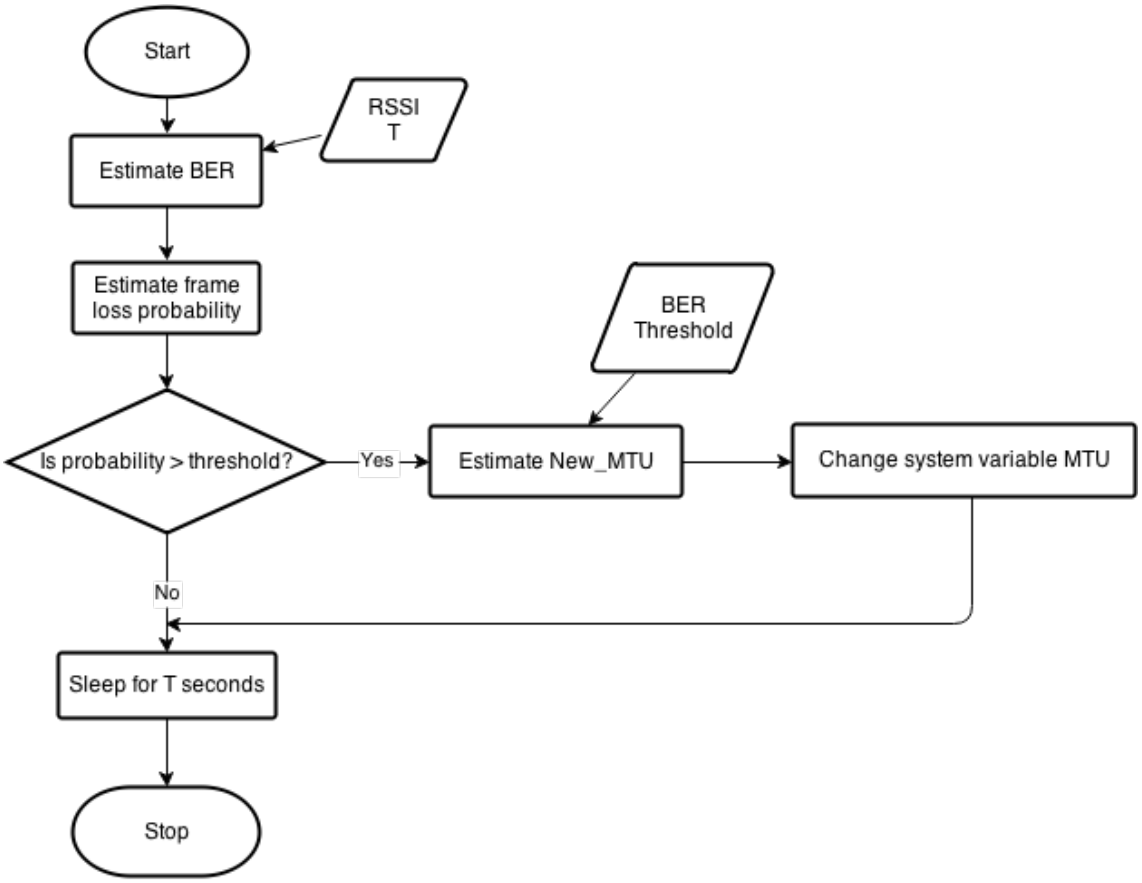

Figure 3.4: Change MTU Algorithm

This last component of the proposed solution changes the Maximum Transmission Unit (MTU) according to the frame loss probability. As referred in Section 2.6, frame loss probability P_L is a

function of collisions and channel error probability P_e. Since our solution is focused on a point-to-point link, frame collision probability can be neglected; so $P_L = P_e$. On the other hand, P_e is calculated according to Bit Error Ratio (BER), related to frame header and payload modulations, and their respective lengths (L), please see Section 2.6. Since the modulations used to transmit the frame header and frame payload are the same, P_L can be defined as:

$$P_L = P_e = 1 - (1 - BER)^L \tag{3.8}$$

The BER formulas, presented in Section 2.7 are different depending on the modulation used. Thus, BER depends on E_b/N_0 which depends on signal noise ratio (SNR), bandwidth and modulation rate. From now on, L is considered to be equal to MTU, since it was proved (using Wireshark [1]) that entire payload is filled with data by traffic generator Iperf (Figure 3.2).

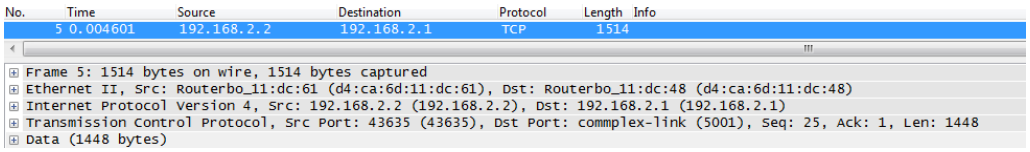

Figure 3.5: Wireshark capture with MTU configured at 1500 bytes

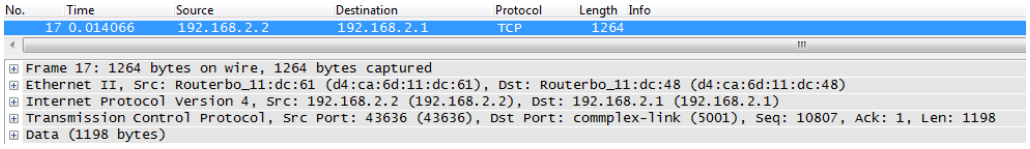

Figure 3.6: Wireshark capture with MTU configured at 1250 bytes

The plots from Figure 3.7 were obtained from a developed algorithm (C language) which retrieves theoretical frame loss probability according to Equation 3.8. In this case, L is fixed at normal 12112 bits (1514 bytes corresponding to MTU 1500), noise at -95 dBm which was the constant measured during all the tests and bandwidth is fixed at 20 MHz. Modulation rate varies from 6.5 Mbit/s (mcs0) to 65.0 Mbit/s (mcs7) using 802.11n.

The final goal of the proposed dynamic MTU algorithm is to keep frame loss probability below a certain threshold. Thus, for a given RSSI, P_L and BER are estimated, if P_L is above the threshold, a new MTU is estimated for given BER, by using the Equation. 3.9. This equation represents a modification of Equation 3.8 in order of L, considering P_e the threshold and L the MTU_{new}:

$$MTU_{new} = log_{1-BER}[1 - threshold] \tag{3.9}$$

[1] http://www.wireshark.org

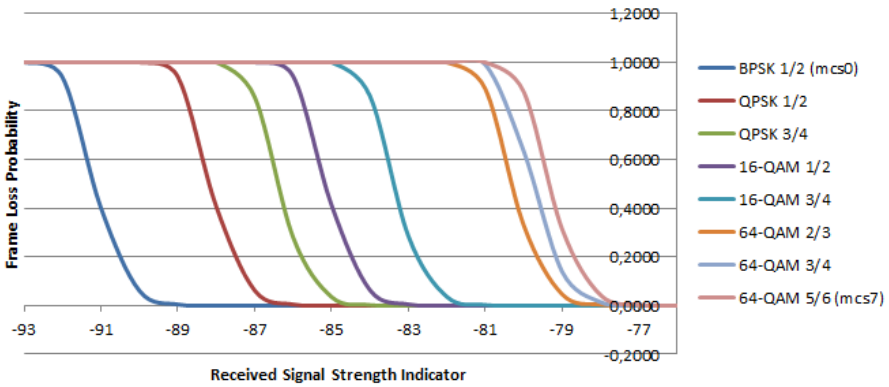

Figure 3.7: 1514 bytes Frame Loss Probability vs signal power (dBm) for mcs0-mcs7

Our implementation in Section 4.2 defines the threshold as well as two approaches used to obtain modulation in use. Thus, it was possible to test the entire proposed solution.

Chapter 4

Proposed Solution Implementation and Evaluation

After presenting the proposed solution in Chapter 3, this chapter introduces the methods used to evaluate, implement and test it. Then, it begins with a comparison between our prediction algorithm and other available in the literature. Also, the procedures for the implementation of the entire algorithm followed by experimental setup, parameters to be considered and test scripts are presented. Finally, the experimental results are discussed and the overall conclusions are drawn.

4.1 Evaluation of the Developed Prediction Algorithm

In order to evaluate our prediction algorithm, it was compared with Grey Model. In Section 2.8 two prediction algorithms entitled Kalman and Grey models were described. Kalman model is an accurate model which require not only knowledge about previous values but also some information of the actual system being used. On the other hand, the Grey model only requires information about previous values. Since RSSI readings were inconclusive about system characteristics and patterns, most commonly used single variable first-order Grey Model was considered, according to [15].

Several tests were performed using both algorithms, using as input the same RSSI values. Those values were obtained in maritime (previous researches) and laboratory RSSI readings from different days. Thus, a one second interval RSSI reader algorithm was also developed.

In order to evaluate an algorithm, on each predicted point, the module of the difference between that value and read one was estimated (error), along with the standard deviation. The plots of fig 4.1 and 4.2 illustrate a sample of an experiment with laboratory measured RSSI during 312 seconds:

The mean from original RSSI for the entire test was -83,2 dBm with a standard deviation of about $2,1$. Maximum series value was -79 dBm and minimum was -88 dBm. Results from this experiment were:

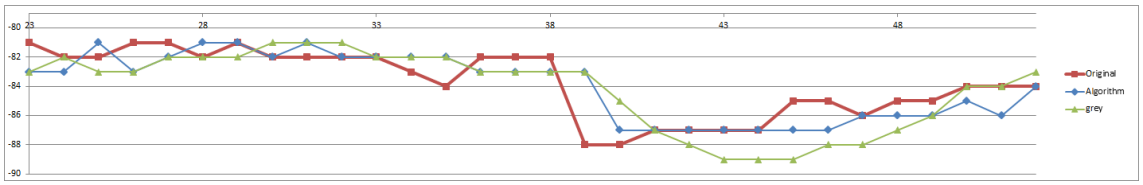

Figure 4.1: Comparison between two prediction algorithms and original signal

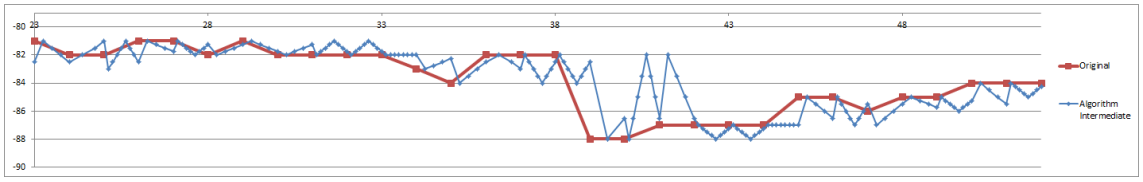

Figure 4.2: Output from developed algorithm and original signal

Our Algorithm vs Grey Model

Method	**Mean Error (dBm)**	**Standard Deviation**
Algorithm	0,96	0,80
Grey Model	1,06	0,88

Table 4.1: Results from experiment one

Next results were obtained from different days and different sea conditions, measured in different days during last year experiments. In the first case, 290 seconds (290 RSSI values) were taken into account, signal varied from -71 to -92 dBm with a mean of -87 dBm and a standard deviation of 2,9:

Our Algorithm vs Grey Model

Method	**Mean Error (dBm)**	**Standard Deviation**
Algorithm	1,62	1,70
Grey Model	1,77	1,96

Table 4.2: Results from experiment two

In the following case, 221 seconds were measured, signal varied from -74 to -89 dBm with a mean of -80,7 dBm and a standard deviation of 3,9:

Our Algorithm vs Grey Model

Method	**Mean Error (dBm)**	**Standard Deviation**
Algorithm	3,52	2,87
Grey Model	3,78	2,64

Table 4.3: Results from experiment three

Although the difference between both algorithms is not significant, the developed algorithm

mean errors were lower than the ones obtained using Grey Model. Thus, these tests can be seen as a validation of the proposed algorithm. In addition, the Grey Model does not predict intermediate values (which are the ones that our algorithm predicts between one second measurements), which would be an obstacle for the proposed solution described in Chapter 3.

4.2 Implementation of the Proposed Solution

This section addresses the implementation necessary to test our solution. As described in Chapter 3, the entire solution culminates with a MTU changing mechanism. MTU changing under OpenWrt (and other OS) is accomplished by using the command:

ifconfig wlan0 mtu x

One of the issues by implementing this solution is the fact that current modulation is a variable hard to obtain. Initially, it was proposed an implementation based on the use of all modulations (Appendix A). It was stated the power wherein each modulation was in use. Those associations were defined taking into account wireless card data sheet and information from contacted manufacturer. This solution was tested with inconclusive results. Real frame loss probabilities were too different from theoretical ones and the improvements were not as good as expected. Again this is due to lack of information about current modulation on 802.11, as not only the RSSI defines it. Thus, it happened that our algorithm was acting like if it was on a modulation when the system was modulation using another rate. This lead to incorrect BER estimation (depends on the bitrate), unexpected, and so non effective, MTU changing.

As such, we fixed the rate in order to draw conclusions on the gains related to dynamic MTU and considering a specific modulation rate. Firstly, it was set to minimum modulation BPSK coding rate: 1/2, mcs0, 6.5 Mbit/s:

iw dev wlan0 set bitrates mcs-5 0

Frame Loss Probability associated to this modulation is the following:

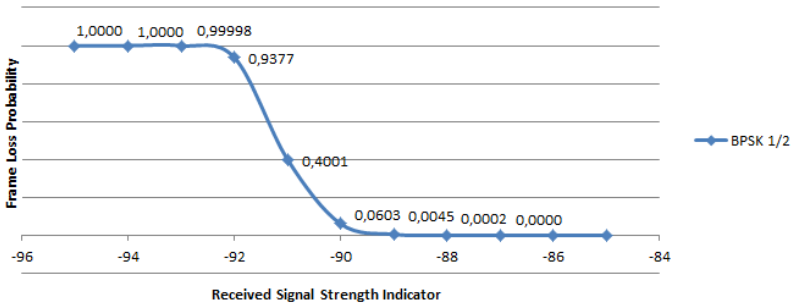

Figure 4.3: Frame loss probability to BPSK at coding rate of 1/2

One implication of using this modulation is the need to test the system with a very week wireless link (near the wireless network interface card sensitivity). This fact lead to several connection losses during laboratory experiments. The problem is that these losses are not detected and the reported RSSI is similar to the one reported when the connection was established. This fact would lead the algorithm to take decisions based on wrong RSSI values.

In order to solve this problem and understand the implications of dynamic MTU, modulation was fixed at 16-QAM with coding rate 3/4 (mcs4, 39 Mbit/s) (Figure 4.4). This modulation was chosen because it allows tests at lower RSSI values (which are the conditions of high distanced nodes). In addition, the values where its frame loss probability equals one are above -85 dBm (Figure 4.4), making RSSI reading possible within all its range. Also mcs4 is the last modulation before 64-QAM and, by analyzing Figure 3.7, it is possible to see that raise from frame loss probability between these modulations is higher. Tests were made for TCP and UDP traffic, considering a saturated wireless link.

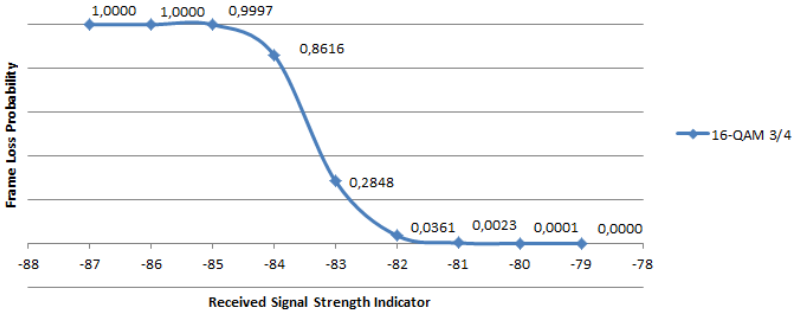

Figure 4.4: Packet loss probability to 16-QAM at coding rate of 3/4

In this case, threshold is defined to be 0.2 since it is slightly lower than frame loss probability at -83 dBm and allows to test MTU changing at RSSI points close to probability zero (which does not require the minimum MTU) and at RSSI points close to one. It is important to note that minimum possible wireless network interface card MTU is 256 bytes. Previous tests lead to the following new MTUs and the plot of Figure 4.5:

RSSI (dBm)	**New MTU (bytes)**	**New frame loss probability**
-83	993	0,20
-84	256	0,30
-85	256	0,77
-86	256	0,99
-87	256	1,00

Table 4.4: New MTU for frame loss probability minimization

Finally, two approaches are considerer in order to change MTU. The first version changes the MTU every time the frame loss probability (calculated according to RSSI value read for the network interface and the prediction provided by our algorithm for the subsequent second) is above

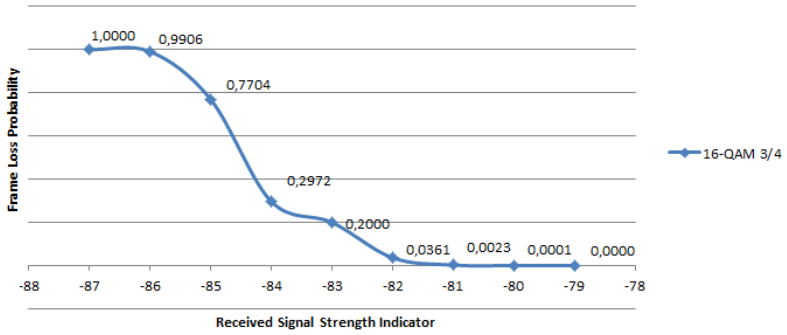

Figure 4.5: New Frame loss probability using the proposed dynamic MTU algorithm

the threshold. This change will last the predicted time interval (T).

The second version firstly runs the entire prediction algorithm in order to understand if the read RSSI value is present in the next second for more than $\frac{2}{3}$ of that period. If yes, the RSSI value is passed to change MTU function and, if frame error probability is above the threshold, the MTU is changed during the entire second.

Next print screen illustrates the second version of the algorithm running under OpenWrt. As can be observed, this version set MTU to 1500 at -82 dBm but did not immediately change MTU to 256 at -86 dBm. On the other hand, if first version was running, the algorithm would be immediately changing the MTU.

```
Mon Jun 16 22:10:43 GMT 2014
RSSI: -82
Default MTU 1500 set
wlan0     Link encap:Ethernet  HWaddr D4:CA:6D:11:DC:48
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25421 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102993 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:32
          RX bytes:1716772 (1.6 MiB)  TX bytes:156564650 (149.3 MiB)

Mon Jun 16 22:10:44 GMT 2014
RSSI: -82
Mon Jun 16 22:10:45 GMT 2014
RSSI: -85
Mon Jun 16 22:10:46 GMT 2014
RSSI: -83
MTU changed to 256
wlan0     Link encap:Ethernet  HWaddr D4:CA:6D:11:DC:48
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:256  Metric:1
          RX packets:25421 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102993 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:32
          RX bytes:1716772 (1.6 MiB)  TX bytes:156564650 (149.3 MiB)

Mon Jun 16 22:10:47 GMT 2014
RSSI: -85
```

Figure 4.6: Dynamic MTU v2 algorithm running under OpenWrt OS

4.3 Experimental Setup

The MARBED testbed was presented in Subsection 2.3.1. The testbed was developed in previous MSc thesis 2.3. Before the experimental evaluation in real environment laboratory tests were performed.

Two systems consisting of Alix3d3 boards with MikroTik R52n-M wireless cards were positioned about one meter from each other and at the same altitude (Figure 4.7). In one of the boards an USB GPS device receiver was connected; even without any signal, it allowed to test the scripts responsible for logging experimental results (later presented). Omnidirectional antennas, having 5 dBm gain, were used in both systems; a 20 dBm attenuator was used in one of them in order to easily reach RSSI values between about -76 to -88 dBm in the laboratory environment. As stated before, these values were required to test the proposed solution.

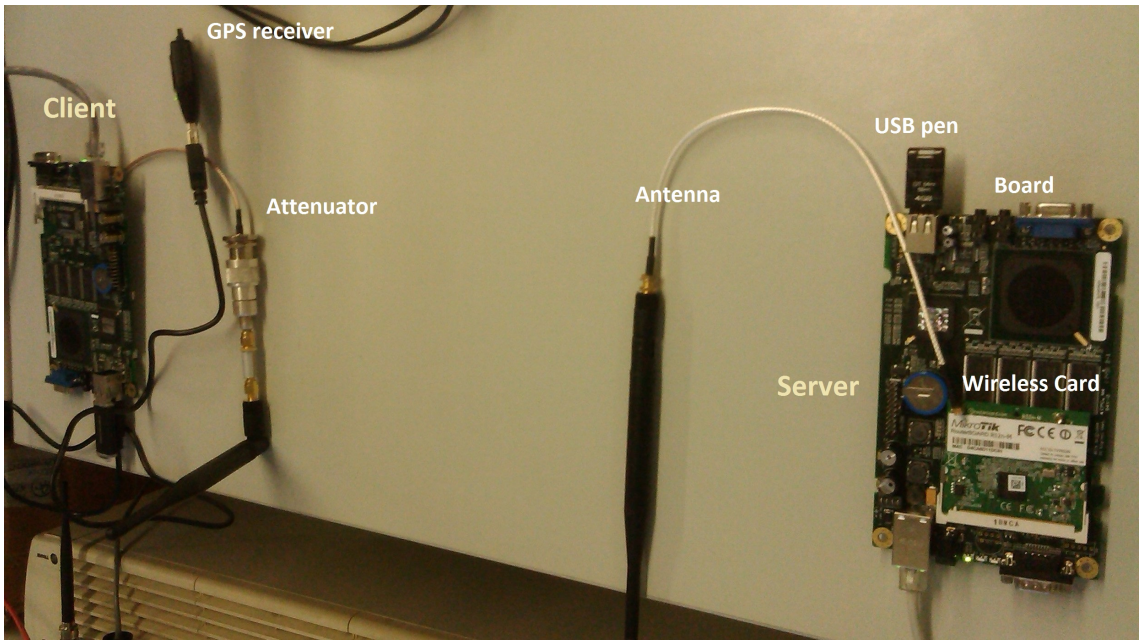

Figure 4.7: Laboratory Testbed

OpenWrt release r36088 named "Attitude Adjustment" (Figure 4.8) was mounted on two USB pen drives. In addition, it was necessary to mount another version suited for development which permitted to debug the algorithm. Along with this version, Eclipse[1] was used to compile and remotely debug. The installation of all applications, namely Iwinfo, Iperf and Gpsd was made making use of *opkg install* command. Th network settings are in Table 4.5.

To perform the tests, one of the stations was defined as the Iperf server and the other as the Iperf client. Three tests ran separately with the server running no algorithm and the client running the first version algorithm, the second version algorithm and no algorithm; when running no algorithm, at the scripts initiation, the MTU was set to 1500.

[1] http://www.eclipse.org/

Parameter	Value
SSID	OpenWrt
Mode	Ad-Hoc 802.11n
Channel	149 (5.745 GHz)
Bitrate	39.0 Mbit/s
Tx-Power	4 dBm (2mW)
HT mode	20 MHz

Table 4.5: Network Specifications

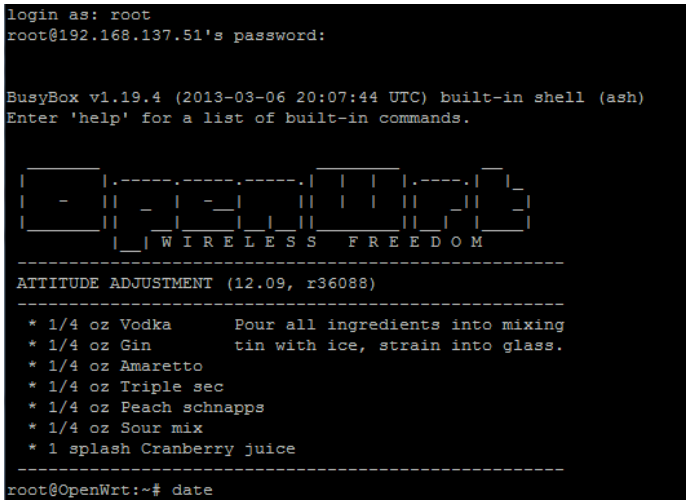

Figure 4.8: Print screen from OpenWrt welcome screen

4.4 Evaluation Metrics

The following set of metrics was selected in order to successfully evaluate the proposed solution:

- **Throughput** - Gives the information, Mbit per second, about link rates (Iperf logs);
- **Jitter** - This parameter measures delay variations (milliseconds) from delivered packets, provided by Iperf uniquely for UDP traffic;
- **RTT** - Measures (in milliseconds) the time it takes for a packet to be delivered on destination, adding the time necessary for the response packet to arrive the origin. This parameter is obtained using ping;
- **RSSI** - Provides signal strength (power in dBm), observed on the mobile node (ship). This parameter is given by RSSI reader algorithm.

4.5 Test Scripts

Shell Scripts were created in order to make the system work autonomously, which is not only a MARBED requirement, but also useful to use and test their robustness in lab testbed. In the

beginning of each script, for safety reasons, modulation was fixed at mcs4 by using the following command:

iw dev wlan0 set bitrates mcs-5 4

Scripts from server and client consist of the following flowcharts (note that all the tasks requiring extra information are signalized with a number being explained above). All processes are initiated in background and terminated using the kill command:

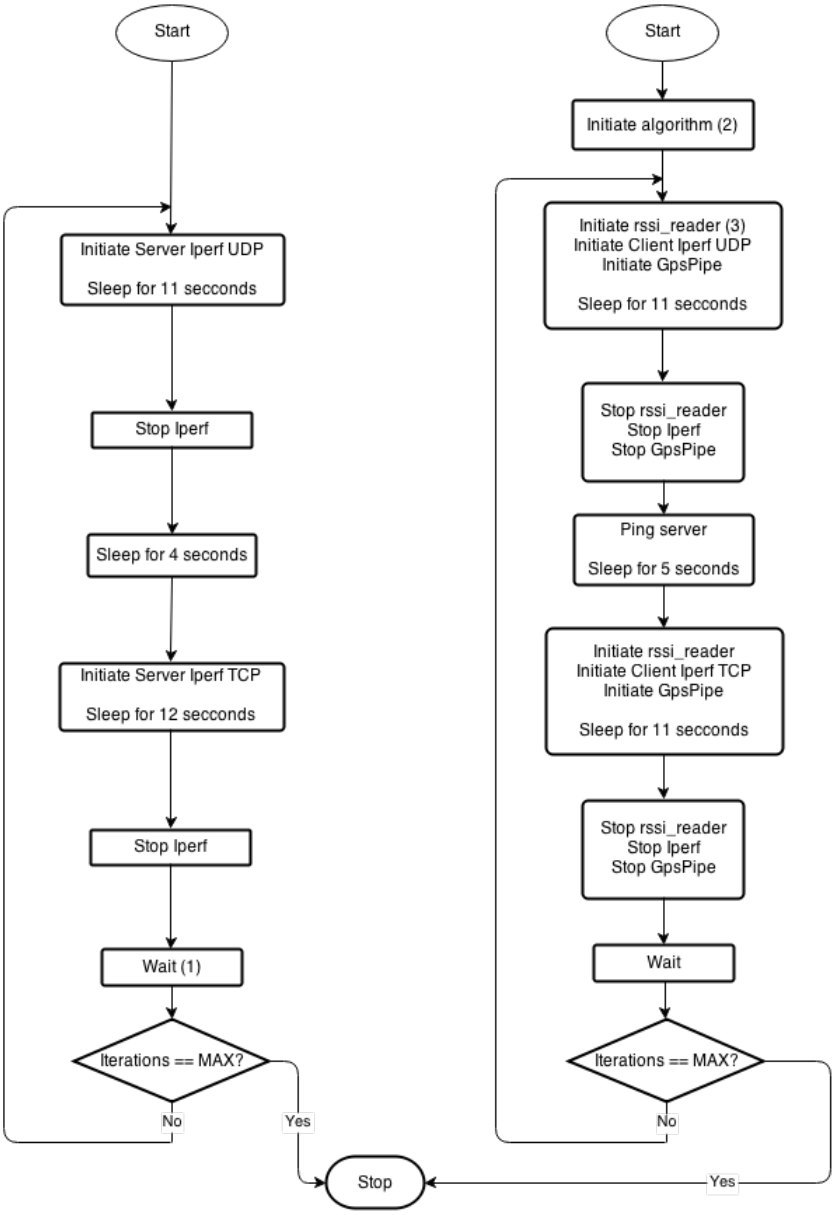

Figure 4.9: Server (left) and client (right) test scripts

- (1) To avoid problems with possible delays from both sides, the algorithms stop in this step until it is second 30 or 00.

- (2) This step is skipped if the no algorithm script is running. If not, there is one algorithm for each version.
- (3) RSSI reader was already mentioned in Chapter 3. Its task is simply to get Rx-power from Iwinfo application every second and log it along with time and date information.

The number of cycle iterations is set when invoking the scripts. Iperf is executed during ten seconds. Ping is ran with 5 iterations of one second each. The output from Iperf UDP, Iperf TCP, RSSI reader UDP, RSSI reader TCP, GpsPipe and ping is saved into different files according to its time, date, mode and source. Next example illustrates the client output for an iteration starting on day 17th at 02 hours, 40 minutes and 00 seconds:

- 17024000.rssi_udp_client; 17024000.rssi_tcp_client
- 17024000.iperf_udp_client; 17024000.iperf_tcp_client
- 17024000.gps_udp; 17024000.gps_tcp
- 17024000.ping

Some issues were found while preparing these tests. One of them was related to time synchronization, even using Crontab[2] to schedule the scripts for the same instant, it was required that the time was exactly the same on both sides. The solution was to run a Network Time Protocol Daemon (NTPD) server on one of the OpenWrt machines and connect to it the NTP client running in the other machine. Another problem was the available space in the USB pen. After running some tests it was observed that logs were no longer being stored. This was due to insufficient storage, since the OpenWrt mounting limits the device to one partition with about 20 free Mbytes. One of the solutions was to create a new partition (ext4) intended to be the path for the logs. Applications required for this tasks were fdisk [3], responsible for partition creation and a set of packages (e2fsprogs [4]) used to mount partition on boot.

4.6 Logs Parsing

In order to parse the logs, C program was created. The program was executed in Linux and its inputs are the logs from Iperf in server side, Ping, Gpspipe and RSSI reader. The output is one file per application containing all the useful data (lines), thus, for Iperf, the lines containing information about throughput and jitter in the server side (effectively received); for Ping, the lines containing the data relative to RTT (ping response); for Gpspipe, the coordinates retrieved every second; finally, for RSSI the lines containing RSSI measurement for each second. Note that these lines were selected by its length (number of characters) and by its position in the file, for example, retrieve the first ten lines counting from the third.

[2] Crontab is an application used for task scheduling available to OpenWrt
[3] http://www.fdisk.com
[4] http://e2fsprogs.sourceforge.net

Logs Parsing Algorithm

```
1:  loop
2:  Open each log file for iteration (day and time)
3:  if Iperf UDP/TCP & ping line_length ∈ (L_min L_max) then
3:      Iperf_line → output_iperf_udp
4:      Rssi_line → output_rssi_udp
5:      Iperf_line → output_iperf_tcp
6:      Rssi_line → output_rssi_tcp
7:      Ping_line → output_ping
8:  else
9:      read next line
10:  end if
12:  if Gps file != null
13:      coordinates → output_gps
14:  end if
15:  end loop
```

L_{max} and L_{min} represent the length bounds for lines containing necessary data. RSSI processing was made according to Iperf logging, since both Iperf and RSSI reader begin simultaneously. For instance, if five lines from Iperf UDP were retrieved (captured by the server), only five lines from RSSI reader UDP log (starting at same time) are send to its output file.

Output processing was made using Microsoft Excel[5] where a table was created containing RSSI and respective parameters (please see Figure 4.10). Also Minitab Statistics[6] was used in order to obtain statistical data such as jitter per RSSI.

	A	B	C	D	E	F	G
1	RSSI	Throughput Static MTU	N		RSSI	Throughput Dynamic MTU	N
2	-86	7,730	5857,000		-79	17,500	9841
3	-84	9,340	RSSI Mean (dBm)		-81	18,600	RSSI Mean (dBm)
4	-84	10,600	-81,58		-81	21,000	-80,77
5	-83	10,200	RSSI StDev		-82	21,800	RSSI StDev
6	-85	11,700	1,95		-81	22,400	3,05
7	-85	11,200	RSSI Median (dBm)		-81	23,500	RSSI Median (dBm)
8	-86	10,000	-81,00		-81	22,900	-80,00
9	-81	12,400	Throughput Mean (Mbits/s)		-81	22,000	Throughput Mean (Mbits/s)
10	-81	15,500	18,61		-80	23,200	21,96
11	-77	25,800	Throughput StDev		-80	19,800	Throughput StDev
12	-78	28,600	5,09		-80	21,800	8,14
13	-77	29,500			-81	22,400	
14	-77	28,900			-81	21,900	
15	-77	27,300			-80	26,700	
16	-77	28,800			-79	28,700	
17	-77	28,500			-78	29,000	
18	-77	30,400			-80	27,300	
19	-78	30,400			-81	26,500	
20	-78	23,500			-80	22,300	
21	-77	28,400			-81	23,700	
22	-78	27,300			-80	28,700	
23	-78	29,000			-80	28,600	
24	-76	28,500			-80	27,800	
25	-78	27,400			-80	28,500	

Figure 4.10: Example of Excel Spreadsheet

[5] office.microsoft.com/pt-pt/excel/
[6] www.minitab.com

4.7 Laboratory Results

The following results were obtained using the lab testbed of Figure 4.7. Several experiments, were performed at different RSSI conditions in order to cover a wider RSSI values range. Tests for higher RSSI values were performed during afternoon and tests for lower RSSI values (where algorithm acts) were scheduled to be done during the night. The tests performed during the night may not be as affected by interference as the performed in the afternoon. However, high RSSI tests had to be done during that time in order to take advantage of the available time. Also, it was crucial to obtain reliable results for low RSSI values so our algorithm effects could be noticed. It is important to refer that prior to any test, or before leaving the laboratory, the antennas were manually positioned while running the RSSI reader in order to assure that results for each solution were obtained at the same conditions. For example, during the night, a test of three hours with no algorithm begun at 23 hours, followed by three hours test with Dynamic MTU algorithm version one followed by same time running version two. This order was changed for next night experiments.

These results refer to statistics from server Iperf, when compared to RSSI values read in the client, necessary to understand the algorithm behavior (in addition, RSSI on client was observed to be quite similar to RSSI on server, since same transmission power was used). All the values in the tables were retrieved by *Minitab*, having as input data such as the presented in Figure 4.10. Each of the samples N represents a one second iteration where client Iperf tries to send as much data as possible, saturating the channel. This is necessary in order to make MTU changes effectively change the packet size.

Subsections 4.7.1, 4.7.2, 4.7.4 and 4.7.3 address results for TCP throughput, UDP throughput, UDP jitter and RTT respectively. Figure 4.11 shows an example of two iterations from server Iperf, first for TCP and then for UDP traffic.

```
------------------------------------------------------------
Server listening on TCP port 5003
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  4] local 192.168.2.2 port 5003 connected with 192.168.2.1
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 1.0 sec  17.0 KBytes   139 Kbits/sec
[  4]  1.0- 2.0 sec  9.90 KBytes  81.1 Kbits/sec
[  4]  2.0- 3.0 sec  11.3 KBytes  92.7 Kbits/sec
[  4]  3.0- 4.0 sec  21.2 KBytes   174 Kbits/sec
[  4]  4.0- 5.0 sec  38.2 KBytes   313 Kbits/sec
[  4]  5.0- 6.0 sec  11.3 KBytes  92.7 Kbits/sec
[  4]  6.0- 7.0 sec  0.00 Bytes  0.00 bits/sec
[  4]  7.0- 8.0 sec  1.41 KBytes  11.6 Kbits/sec
[  4]  8.0- 9.0 sec  11.3 KBytes  92.7 Kbits/sec
[  4]  9.0-10.0 sec  0.00 Bytes  0.00 bits/sec
```

```
------------------------------------------------------------
Server listening on UDP port 5002
Receiving 1470 byte datagrams
UDP buffer size:  160 KByte (default)
------------------------------------------------------------
[  3] local 192.168.2.2 port 5002 connected with 192.168.2.1
[ ID] Interval       Transfer     Bandwidth        Jitter
[  3]  0.0- 1.0 sec   128 KBytes  1.05 Mbits/sec   6.709 ms
[  3]  1.0- 2.0 sec   158 KBytes  1.29 Mbits/sec   9.367 ms
[  3]  2.0- 3.0 sec   187 KBytes  1.53 Mbits/sec  10.335 ms
[  3]  3.0- 4.0 sec   155 KBytes  1.27 Mbits/sec   7.843 ms
[  3]  4.0- 5.0 sec   148 KBytes  1.21 Mbits/sec  15.830 ms
[  3]  5.0- 6.0 sec   146 KBytes  1.20 Mbits/sec  13.521 ms
[  3]  6.0- 7.0 sec   144 KBytes  1.18 Mbits/sec   9.825 ms
[  3]  7.0- 8.0 sec   116 KBytes   953 Kbits/sec  11.129 ms
[  3]  8.0- 9.0 sec   86.1 KBytes   706 Kbits/sec  23.516 ms
[  3]  9.0-10.0 sec  83.3 KBytes   682 Kbits/sec  20.882 ms
```

Figure 4.11: Server Iperf example of TCP and UDP capture

4.7.1 TCP Throughput

	Dynamic MTU v2			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-82	294	0,18	0,96	0,05
-81	948	0,21	0,85	0,10
-80	585	0,20	0,23	0,15

Table 4.6: Throughput (Mbit/s) for Dynamic MTU v2 using TCP

After the first tests, it was concluded that the second version of the algorithm did not demonstrate improvements. Thus, it was not tested for as long as first version. This explains the larger number of samples present in those tests and the wider RSSI range presented.

	Static MTU				Dynamic MTU v1			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-82	250	0,22	1,06	0,00	284	0,53	2,09	0,09
-81	1050	0,31	1,05	0,07	1347	0,39	1,57	0,16
-80	1011	0,43	0,84	0,27	1491	0,36	1,04	0,20
-79	796	11,04	9,92	11,80	1250	10,05	9,90	6,43
-78	507	21,00	6,30	23,40	430	19,24	9,05	23,90
-77	465	24,11	3,83	25,50	534	25,06	3,71	26,10
-76	389	25,57	1,78	26,00	704	25,96	1,38	26,30
-75	159	25,93	1,23	26,20	902	26,10	1,65	26,50
-74	53	25,83	1,51	26,20	480	26,15	1,75	26,50

Table 4.7: Throughput (Mbit/s) for Dynamic and Static MTU v1 using TCP

Table 4.6 shows that version 2 of dynamic MTU algorithm does not present any improvement when compared to static MTU. From this, we can conclude that it is preferable to change MTU once the RSSI drops, instead of changing when it is predicted to be constant.

Against theoretical model for mcs4, any of the tests with RSSI measures below -82 dBm showed TCP traffic statistics, indicating 100% packet loss. This can be due to harsh RSSI oscillations in low signal strength during the interval where no reading are obtained from the wireless card.

Table 4.8 evidences the differences from both systems. Gains were estimated using the formula

$$gain(\%) = \frac{(variable_{dynamic} - variable_{static})}{variable_{static}} * 100 \quad (4.1)$$

It is relevant to note that, at -82 dBm, the coverage gain from Dynamic MTU is about 136 % and it is not possible to estimate median gain since median for static MTU throughput was 0 Mbit/s. This can be explained through the prediction algorithm. Even being the RSSI reader retrieving these real values, the algorithm may predict lower values during the one second interval, leading to an MTU decrease and, consequently to higher throughput due to lower packet loss and fewer retransmissions. Standard deviation also helps proving this explanation since it is almost twice when

Gains from Dynamic MTU regarding Static MTU (throughput)

RSSI (dBm)	Mean Gain (%)	StDev Gain (%)	Median Gain (%)
-82	**136**	97	-
-81	**24**	50	**133**
-80	-18	24	-26
-79	-9	-0,2	-46
-78	-8	44	2
-77	4	-3	2
-76	2	-22	1
-75	1	34	1
-74	1	16	1

Table 4.8: Comparison between Dynamic and Static MTU systems (Throughput TCP)

compared to static MTU; for the same reason, an improvement of around 24% on mean throughput and 133% on median was possible at -81 dBm. We consider that worst results, observed using the algorithm, for RSSI -80, -79 and -78 dBm are explained by wrong RSSI predictions after these readings. Thus, if the algorithm reads these values and predicts a variation having as consequence MTU changing (above -82 dBm), it is shown than that it is not compensatory.

The plot from Figure 4.12 shows the global results while plot from Figure 4.13 shows zoomed results for low RSSI:

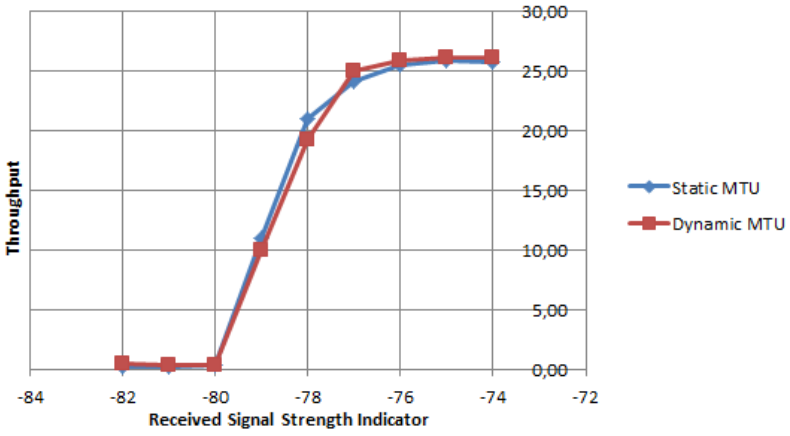

Figure 4.12: TCP mean Throughput (Mbit/s) according to global RSSI (dBm)

Analysis of the entire sample disaggregated from RSSI is presented in the Table 4.9 and in the Cumulative Distribution Function (CDF) from Figure 4.14 obtained using Minitab. Based on those values, version one of our algorithm shows a TCP throughput gain of 20,76%.

This gain refers to the entire sample with no signal strength distinctions which is not reliable since the link characteristics oscillated differently, for instance, it can occur in a test that RSSI is -80 dBm for most of the time and in the other test, half an hour at -82 dBm. This introduces

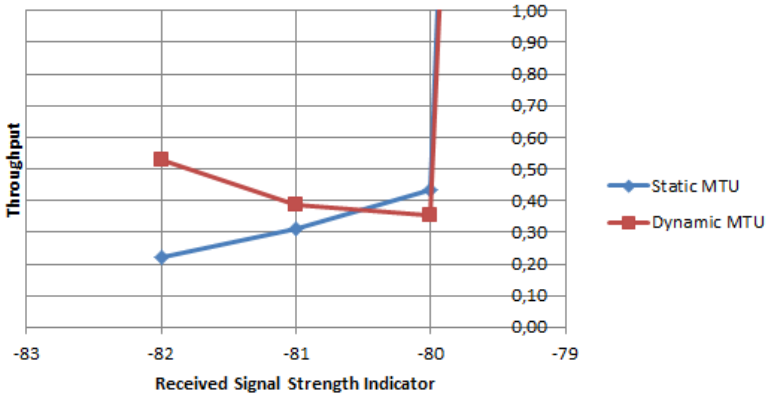

Figure 4.13: TCP mean Throughput (Mbit/s) according to low RSSI (dBm)

Static MTU			**Dynamic MTU v1**		
N	**Mean**	**StDev**	**N**	**Mean**	**StDev**
4680	10,02	11,57	7422	12,10	12,33

Table 4.9: Throughput (Mbit/s) for the entire sample

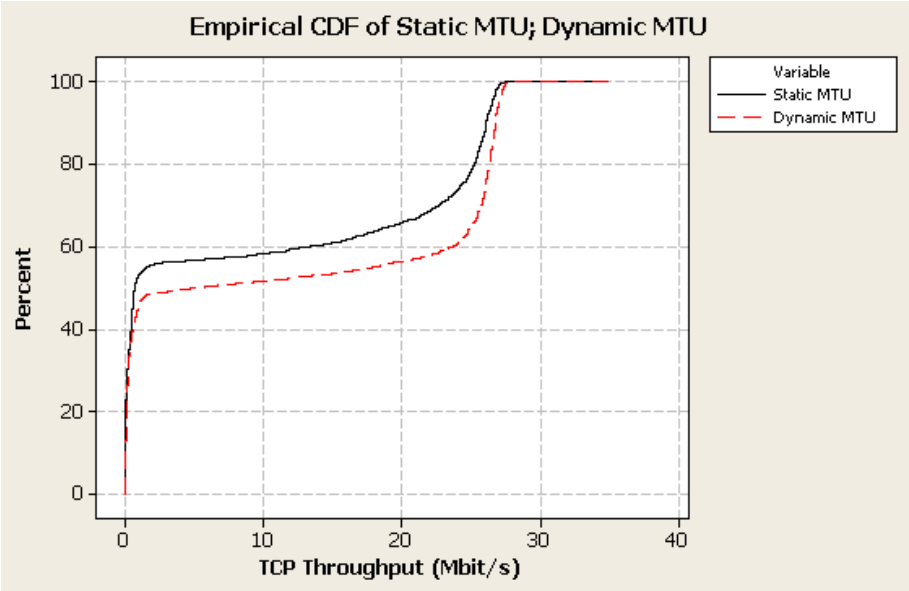

Figure 4.14: CDF for TCP throughput (Mbit/s) for Dynamic and Static MTU

considerable changes when analyzing the entire sample, so the samples were discriminated per RSSI.

CDF main conclusion is that TCP throughput is higher for dynamic MTU in the most of the samples. This can be observed by observing that, for the same percentage (above about 30%), throughput has a higher value on the X axis. For example, 80% of the values are below 25 Mbit/s using static MTU, different from dynamic MTU, where the same percentage of values are below

about 32 Mbit/s. Showing an increase of about 28% in a considerable part of the sample.

Number of samples are bigger for Dynamic MTU indicating that the Iperf server captured more frames in this case. This fact was not addressed and was not statistically analyzed because it was caused by total connection loss in certain instants which would result in same losses if the algorithm was being used, not reflecting its any improvements.

4.7.2 UDP Throughput

	Dynamic MTU v2			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-83	301	0,17	0,36	0,08
-82	2307	0,27	0,42	0,11
-81	2802	0,41	0,55	0,13
-80	769	0,64	0,63	0,40

Table 4.10: Throughput (Mbit/s) for Dynamic MTU v2 using UDP

As stated before, there are no significant improvements from version 2 of Dynamic MTU and Static MTU, justifying the lower number of samples gathered by testing this version. This also explains the fact that the RSSI values for v2 tests are in lower number, since there were no samples for other values.

	Static MTU				**Dynamic MTU v1**			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-83	209	0,15	0,22	0,11	262	0,22	0,43	0,11
-82	1570	0,27	0,41	0,13	1891	0,37	0,56	0,12
-81	2645	0,54	0,73	0,28	3021	0,91	1,61	0,66
-80	1546	2,01	4,71	0,73	1731	2,30	4,50	1,41
-79	804	14,44	12,10	14,00	645	8,89	10,79	1,99
-78	490	26,72	7,22	29,65	855	26,01	8,69	30,10
-77	518	29,97	4,16	31,60	497	30,98	3,58	32,00
-76	319	31,38	2,42	32,00	808	31,45	2,46	32,00
-75	160	31,62	1,81	32,00	810	31,50	2,84	32,00

Table 4.11: Throughput (Mbit/s) for Dynamic and Static MTU v1 using UDP

Table 4.11 shows the measurements for throughput according to the read RSSI at its measurement instant. We can conclude that mean throughput is higher than for TCP traffic, this occurs due to no use of acknowledgment (ACK). In TCP, it may happen that a frame is successfully delivered, being the ACK not received by the client. This may lead to wrong retransmissions, reducing throughput.

These results allow concluding that UDP traffic is possible at lower link power when compared to TCP (minimum RSSI with Iperf was -82 against UDP -83 dBm). This occurs due to the Iperf requirement of establishing the TCP connection in the beginning of its execution. If this connection

is not successful, meaning server is not reached by the client due to low RSSI (no ACK received), client does not retry to connect and the entire low RSSI test is lost. On the other hand, UDP does not require a connection and the client always sends data each second even if no ACK is received, this allows to have successful iterations at lower RSSI values. Medium throughput for stable link, above -76 dBm, presents throughputs of about 32 Mbit/sec, proving that the channel is saturated at this limit.

Gains from Dynamic MTU regarding Static MTU (Throughput)

RSSI (dBm)	Mean Gain (%)	StDev Gain (%)	Median Gain (%)
-83	**43**	**93**	**0**
-82	**37**	**37**	**-9**
-81	**68**	**120**	**134**
-80	**14**	-4	**93**
-79	-38	-11	-86
-78	-3	20	2
-77	3	-14	1
-76	0,2	2	0
-75	-0,4	57	0

Table 4.12: Comparison between Dynamic and Static MTU systems (Throughput UDP)

The results shown in table also 4.12 demonstrate the conclusions drawn for TCP traffic. Again, we can see that, at lower RSSI, the throughput gains are bigger because the algorithm is triggered, making changes in MTU. Side effects of the algorithm are again demonstrated by standard deviation which is almost double (-83 dBm) when compared to static MTU. This can be related to the moments when predicted RSSI values are different than the read, leading to MTU changing. For instance, if -83 dBm value is read, MTU is changed to 993 bytes, leading to higher throughputs. Then, if it is predicted a RSSI value of -82 dBm, the MTU returns to 1500 bytes changing the throughput, all for the same read RSSI -83 dBm.

As the RSSI increases, the gain decreases, except at -79 dBm where the throughput drop almost 40% (Table 4.12). This value might have happened due to uncontrollable factors such as interference or unexpected movement, since those tests were realized in the afternoon and the algorithm influence should be close to null.

The plot of Figure 4.15 aims to give a different perspective of the results, being the plot of Figure 4.16 used to reinforce the differences for low RSSI results.

Analyzing throughput for the entire sample (Table 4.13 and CDF from Figure 4.17), the estimated gain is about 32,54%. This values represent the entire sample but are not conclusive due to different variations in RSSI in the diverse tests. For this reason, the analysis per RSSI was elaborated.

UDP global results, when compared to TCP show an unexpected throughput mean decrease, not being consistent with throughputs per RSSI from tables 4.7 and 4.11 which are bigger for UDP traffic. This is due to UDP test conditions which is were proved to be mostly worst (lower RSSI) when compared to TCP; reinforcing the need to discriminate results for RSSI

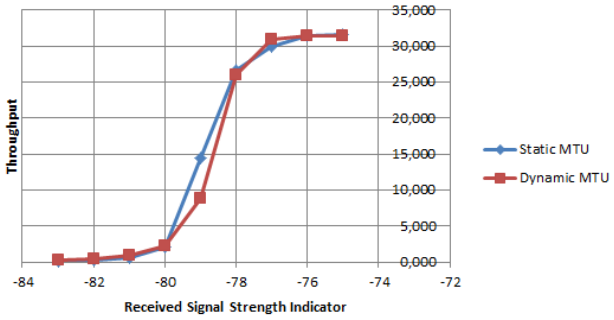

Figure 4.15: UDP mean Throughput (Mbit/s) as a function of RSSI (dBm)

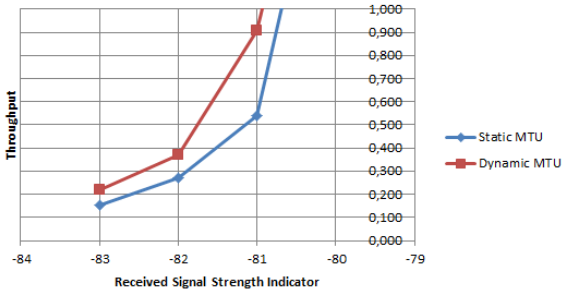

Figure 4.16: UDP mean Throughput (Mbit/s) according to low RSSI (dBm)

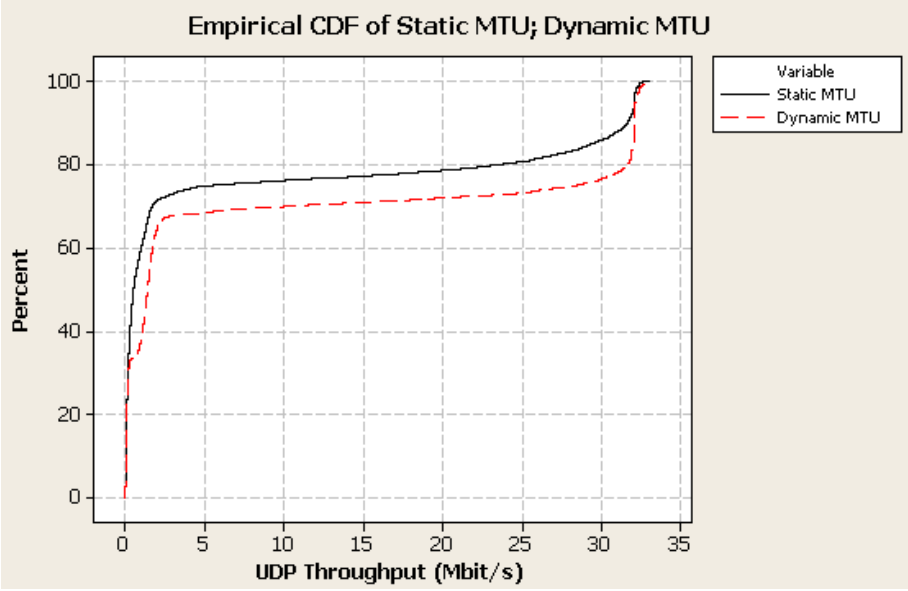

Figure 4.17: CDF for UDP throughput (Mbit/s) for Dynamic and Static MTU

By analyzing CDF, again it remains clear the improvements brought by dynamic MTU, which reveals higher throughputs for the most of the samples. In this case, using static MTU, about 80% of the samples had a throughput lower than 23 Mbit/s, being about 32 Mbit/s for dynamic MTU.

Static MTU			**Dynamic MTU v1**		
N	**Mean**	**StDev**	**N**	**Mean**	**StDev**
8261	7,30	12,12	10520	9,67	13,57

Table 4.13: Throughput (Mbit/s) for the entire sample

Dynamic MTU gain for 80% of the values is about 32%.

Once again the number of samples observed in server Iperf using static MTU is inferior to Dynamic MTU (about 27%). As explained for TCP before, this reveals total uncontrolled connection loss periods during the time where static MTU tests were being performed and would not be avoided by dynamic MTU algorithm. So, the amount of these uncontrollable connection losses was lower in the period of dynamic MTU tests.

4.7.3 UDP Jitter

	Dynamic MTU v2			
RSSI	**N**	**Mean**	**StDev**	**Median**
-83	301	76,79	34,97	75,90
-82	2307	67,07	36,83	68,37
-81	2802	58,08	38,65	57,33
-80	769	42,19	36,14	25,07

Table 4.14: Jitter (ms) for Dynamic MTU v2 using UDP

Jitter results for version 2 of dynamic MTU are similar to version 1 and static MTU as showed above.

	Static MTU				**Dynamic MTU v1**			
RSSI	**N**	**Mean**	**StDev**	**Median**	**N**	**Mean**	**StDev**	**Median**
-83	209	78,96	41,75	73,72	26	82,56	46,60	77,67
-82	1570	68,69	42,05	65,18	1891	71,15	57,91	68,25
-81	2645	48,61	38,25	39,66	3021	45,68	45,03	22,72
-80	1546	30,32	32,00	19,57	1731	20,75	28,85	11,70
-79	804	8,27	13,21	1,21	645	7,62	8,46	6,85
-78	490	0,86	2,80	0,27	855	1,08	3,11	0,28
-77	518	0,35	0,92	0,17	497	0,22	0,33	0,12
-76	319	0,22	0,43	0,13	808	0,16	0,12	0,11
-75	160	0,16	0,22	0,13	810	0,15	0,11	0,12

Table 4.15: Jitter (ms) for Dynamic and Static MTU v1 using UDP

Main conclusions relative to UDP traffic were presented on previous section but looking at Table 4.16, we can conclude that there is no visible influence on jitter by the Dynamic MTU algorithm. As expected, by observing table 4.15, higher jitter measurements occurred at lower RSSI values.

Gains from Dynamic MTU regarding Static MTU (Jitter)

RSSI	Mean Gain (%)	StDev Gain (%)	Median Gain (%)
-83	5	12	5
-82	4	38	5
-81	-6	18	-43
-80	-32	-10	-40
-79	-8	-36	466
-78	25	11	3
-77	-39	-64	-26
-76	-27	-72	-15
-75	-8	-50	-7

Table 4.16: Comparison between Dynamic and Static MTU systems (Jitter UDP)

Representing the difference between delays, jitter increases for worst links. In this cases delays are introduced and so, packet arrival pace is not constant.

The plot of Figure 4.18 graphically shows jitter evolution according to RSSI, where there are no relevant discrepancies to highlight.

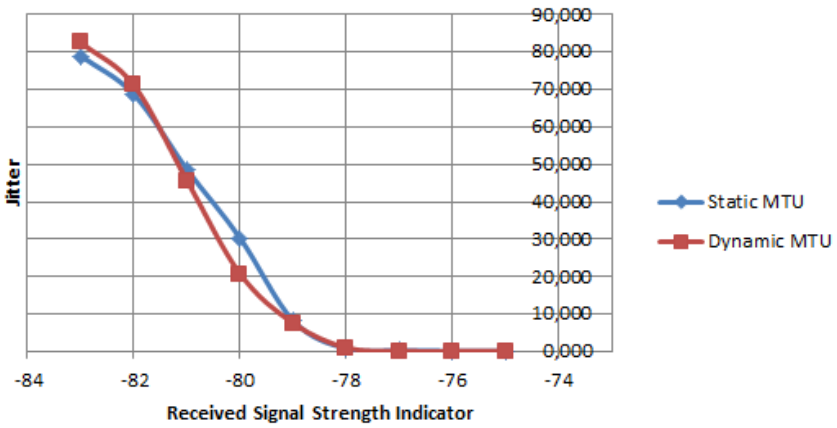

Figure 4.18: UDP mean Jitter (*ms*) according to RSSI (dBm)

4.7.4 Round-Trip-Time

Regarding round-trip-time, the results are presented in Table 4.17. These measurements were obtained using ping command provided by OpenWrt OS. Ping functionality sends a 64 bytes packet and waits until the receive of the same data from its destination, being retrieved RTT which is the time it took for the packet to go and come back. In this case, each sample N represents a request-response iteration.

If we look to mean RTT values, a 25,4% gain is observed when using the proposed dynamic MTU. CDF for RTT is presented in the plot from Figure 4.19. On the other hand, the gains from Median are not significant, 0,7%.

Method	N	Mean	StDev	Median
Static MTU	2578	96,86	194,85	10,30
Dynamic MTU	2829	72,25	171,13	10,38
Gain (%)	-	-25	-12	0,7

Table 4.17: Ping RTT (ms)

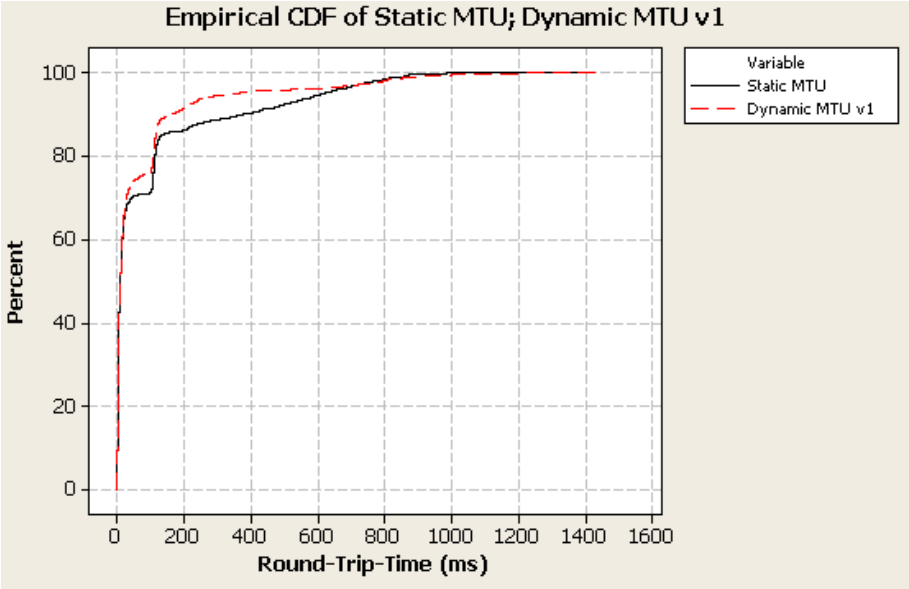

Figure 4.19: CDF for RTT (ms) for Dynamic and Static MTU

Observation from CDF allows understanding that RTT differences are negligible, meaning that the results for RTT mean (Table 4.17) are higher for static MTU due to a low amount of samples which could be taken from periods where the RSSI was too low. Those low RSSI conditions were not replicated during dynamic MTU experiments, so the mean was not affected.

Conclusions from median and from CDF allow to state that algorithm does not influence Round-Trip-Time.

4.8 Maritime Testbed Results

The next step would be to test the proposed dynamic MTU solution in maritime environment using the MARBED testbed described in Section 2.3.

Due to few available time, only during two weeks our proposed solution could be tested using MARBED. The ship just follows its normal fishing route and, unfortunately, during this period, ten working days, no route was in south direction (please see Figure 4.20) to enable communications between the sea node and the land station. For this reason, it was not possible to obtain any results in this testbed.

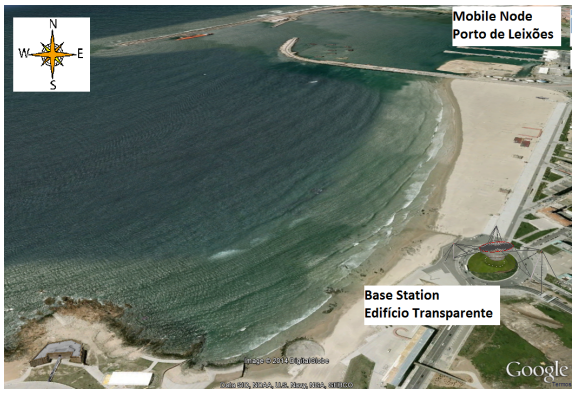

Figure 4.20: Maritime Scenario

4.8.1 Robotics Exercise 2014 (REX14) - "Estuário do Tejo"

On the other hand we were able to participate in this event promoted by the Portuguese Navy Army called REX14. Here we were able to test our solution not only using moving ships but also with steady ships. In the following sections the results are the ones obtained using the steady ships, since the tests with moving nodes shew too few results in order to be conclusive.

The system used in these tests was the same as the used in the laboratory ones only without the attenuator. The ships was the client and a land station was the server as can be observed in figure 4.21.

Figure 4.21: Maritime Testbed

4.8.2 TCP Throughput

Again, as observed in the laboratory results, our dynamic MTU algorithm shew improvements of more than the double for lower RSSI regimes, being these gains decreasing as the RSSI is increasing and our algorithm does not actuate.

	Static MTU				Dynamic MTU v1			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-82	24	2,38	4,67	0,33	25	5,38	6,71	4,16
-81	61	3,78	5,12	1,49	71	8,37	7,81	7,29
-80	80	7,71	7,07	6,13	127	12,84	7,96	14,30
-79	95	12,64	8,49	13,90	143	18,52	7,15	20,30
-78	55	16,48	6,87	16,50	132	21,84	5,24	23,60
-77	18	21,74	6,31	24,95	94	23,32	4,58	25,15
-76	5	25,28	1,31	25,20	48	25,65	1,76	26,25

Table 4.18: Throughput (Mbit/s) for Dynamic and Static MTU v1 using TCP

Gains from Dynamic MTU regarding Static MTU (throughput)

RSSI (dBm)	Mean Gain (%)	StDev Gain (%)	Median Gain (%)
-82	**126**	44	-
-81	**121**	52	389
-80	66	13	133
-79	47	-16	46
-78	33	-24	43
-77	7	-28	0,8
-76	1	34	4

Table 4.19: Comparison between Dynamic and Static MTU systems (Throughput TCP)

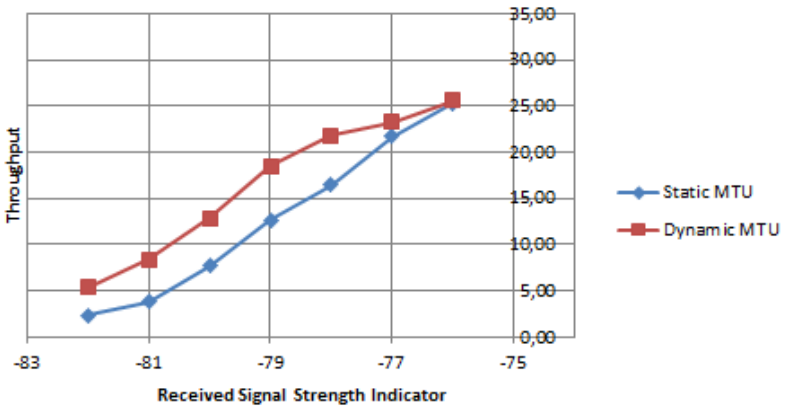

Figure 4.22: TCP mean Throughput (Mbit/s) according to global RSSI (dBm)

Throughput curves are the ones shown in Figure 4.22. We can observe that these curves differ from the throughput curves obtained in laboratory tests. This is due to the smaller amount of samples obtained in REX14, due to our time limitations.

4.8.3 UDP Throughput

As for TCP throughput, we also observe gains in UDP throughput in maritime environment for low RSSI regimes. Lower gains when compared to TCP are again related to the fact that in TCP,

	Static MTU				Dynamic MTU v1			
RSSI (dBm)	**N**	**Mean (Mbit/s)**	**StDev**	**Median**	**N**	**Mean (Mbit/s)**	**StDev**	**Median**
-83	70	1,91	4,58	0,07	24	3,61	5,56	0,95
-82	87	2,42	4,55	0,27	45	3,59	5,23	1,75
-81	127	4,85	6,44	2,15	80	8,84	8,61	5,83
-80	145	10,01	9,14	7,94	159	15,13	10,12	14,70
-79	104	15,11	10,27	13,85	152	22,74	8,56	25,10
-78	48	23,44	8,51	26,10	580	27,20	6,74	29,80
-77	14	28,19	5,93	30,65	100	31,07	2,63	31,90
-76	7	29,30	3,57	30,50	52	31,01	3,62	32,00

Table 4.20: Throughput (Mbit/s) for Dynamic and Static MTU v1 using UDP

Gains from Dynamic MTU regarding Static MTU (Throughput)			
RSSI (dBm)	**Mean Gain (%)**	**StDev Gain (%)**	**Median Gain (%)**
-83	**89**	21	1238
-82	**48**	15	548
-81	**82**	34	171
-80	51	10	85
-79	51	-16	81
-78	16	-20	14
-77	10	-55	4
-76	6	1	5

Table 4.21: Comparison between Dynamic and Static MTU systems (Throughput UDP)

if a packet is lost, there will be a retransmission from ACK no receive. Thus, if packet loss is inferior, the gains will be more representative than for UDP.

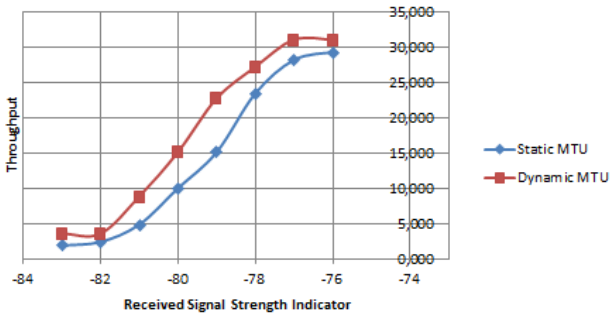

Figure 4.23: UDP mean Throughput (Mbit/s) as a function of RSSI (dBm)

In Figure 4.23, we can see that throughput maritime curves approximate from the ones obtained in laboratory environment, in spite of the lower number of samples, meaning the obtained samples can be seen as statistically relevant.

4.9 Conclusions and Discussion

In this chapter we presented the effects of using the proposed Dynamic MTU algorithm. It can be considered especially valid and efficient in poor link quality conditions networks as it would be expected. Globally, better throughputs were achieved at low RSSI values where MTU changing occurs; higher gains were found for TCP. On the other hand, increasing on standard deviation were obtained, which may lead to network instability.

These results are also a motivation to take this mechanism to adaptive modulations. Our researches in this area lead to inconclusive results due to its variability from one test to other. This happens due to uncertainty about the current modulation and so, wrong packet error probability estimation making decrease MTU when it not necessary. Also, since adaptive modulation mechanism is not synchronized with the dynamic MTU proposed solution, it may happen, for instance, that our algorithm lowers the MTU while the modulation was already adapted in order to reduce BER. Thus, our algorithm assumptions are, actually, not verified. Results from this implementation can be seen in Appendix A.

The target of this research was achieved, since we improved channel characteristics at low RSSI links, such as the conditions found in the maritime environment. In spite of this solution was not tested in the sea, greater distances (lower RSSI) can be reached within an acceptable frame loss probability.

Although packet loss is an interesting parameter to be considered, its analysis was considered impracticable. This parameter was initially obtained using Iperf logs which are not reliable since their information is based in the loss relative to the application generated traffic, 32 Mbit/s user defined. Ping losses with 1500 bytes packets were considered in alternative. This approach was tested while the channel was free and while the channel was saturated, showing only ping replies at higher RSSI values (free channel) and no replies at all with saturated channel. We concluded these results are related to the ping response which has the same length as ping request, leading to extremely high loss probability. Still, the results derived from other parameters analysis allow to induce that lower frame loss ratio is being achieved and that this fact leads to higher throughput along with all respective consequences.

The Source Code of all presented algorithms, OpenWrt executable files and Scripts are available in a package through the link: http://tinyurl.com/mpv8rej.

Chapter 5

Conclusions and Future Work

Results from this work allow to conclude that dynamic MTU improves poor Wi-Fi links. The proposed RSSI prediction algorithm was one of the contributions from this dissertation showing lower errors when compared with state of the art counterparts and bringing up a new method of estimating the values of a given input variable between two samples taken in a period. In some way, it allows to transform a discrete function in a continuous one and is a method not limited to the RSSI variable targeted in this work.

On the other hand, it was proved that when the signal strength of a link is low, the reduction of the MTU results in the minimization of frame error ratio. This fact was introduced by the theoretical model and was proved with experimental results. For both TCP and UDP traffic, gains were achieved. For TCP, dynamic MTU showed a maximum gain of 133% while, for UDP, a maximum gain of 68% was achieved. Also, it was verified that jitter and RTT are not significantly improved by the proposed solution. A tradeoff of this approach is the high throughput variability represented by standard deviation. Since higher throughputs were observed, this characteristic may be seen as an interesting factor because it means that the RSSI predictions were being successful. Thus, fast MTU changes were occurring leading to high variations in the observed throughput. Although our experiments with adaptive modulation were not successful due to the issue of finding out the modulation currently in use, the results indicate that, if this algorithm works together and synchronized with the adaptive modulation one, further gains can be achieved, in particular for low RSSI regimes.

The maritime environment has specific characteristics leading to low and unstable RSSI. The proposed solution was designed in order to address such characteristics. In spite of MARBED tests were not successful due to the routes followed by the fishing ship used as the sea node we had the chance to test our solution in real conditions in REX14. Laboratory experimental results allow concluding that, in these conditions, better throughputs and longer ranges, due to the possibility of communicating with stable throughputs in link conditions where static MTU does not provide them, could be a consequence of implementing our proposed solution; Field experimental results also support this conclusion, showing closer gains to the ones obtained in laboratory validating our solution for a real case scenario.

The development of this MSc thesis work brought up a set of new other researches and improvements that can be based on our proposed solution:

- **Large scale implementation** - We highlight the need to associate our algorithm with the adaptive modulation in order to make this solution a global MAC amendment of wide and easy implementation.
- **RSSI announcement** - Other improvement to this algorithm would consist in making possible to work with RSSI measured at its destination. This could bring more reliable conclusions since, in this work, we assumed an equal RSSI in both terminals. In addition this could lead to a new MAC where all the stations in a network announce their RSSI information to the correspondent node, allowing the algorithm to be running in all the stations within it.
- **Multiple nodes network** - Future work related to our proposed solution could consist of using point-to-multipoint and multipoint-to-point Wi-Fi communications. This could be achieved using three or more nodes. It is expected that, by reducing MTU, collision probability also is reduced, bringing improvements to these scenarios.
- **Relay solution** - Another method which could be used to improve maritime connections is the use of relay servers. This could be achieved either through the use of floating stations or through mesh networks comprised by mobile sea nodes (ships).

Appendix A

Adaptive Modulation Experiments

This appendix refers to experiments performed with the Wi-Fi running in adaptive modulation regime. These tests include TCP throughput and Round-Trip-Time. Regarding this issue, experimental setup, results and conclusions are presented.

A.1 Experimental Setup

In order to perform these experiments, it was assumed that modulations vary according to RSSI values shown in Table A.1:

RSSI (dBm)	Modulation	Bitrate (Mbit/s)
-95	BPSK R=$\frac{1}{2}$	6,5
-87	QPSK R=$\frac{1}{2}$	13
-85	QPSK R=$\frac{3}{4}$	19,5
-84	16-QAM R=$\frac{1}{2}$	26
-82	16-QAM R=$\frac{3}{4}$	39
-79	64-QAM R=$\frac{2}{3}$	52
-78	64-QAM R=$\frac{3}{4}$	58,5
-77	64-QAM R=$\frac{5}{6}$	65

Table A.1: Modulation according to RSSI

Please note that there are no values below -95 dBm, which is the wireless card sensitivity, and above -77 which is the value where maximum bitrate is enabled. Assuming this modulation scheme, the Frame Loss Probability as a function of the RSSI with adaptive modulation is shown in Figure A.1:

In this experiments, the frame loss probability threshold was set to 0,0058. This value was chosen due to the presence of three RSSI values in which it was exceeded. This limit would allow the MTU to be always changed to values above the wireless card minimum (256 bytes) and so to be always above that threshold as can be seen in Table A.2.

These modifications lead to the comparison represented in the plot of Figure A.2:

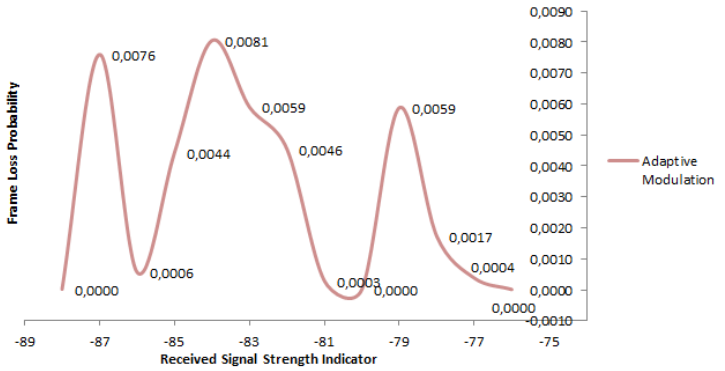

Figure A.1: 1514 bytes Frame Loss Probability vs signal (dBm) for Adaptive modulation

RSSI (dBm)	New MTU (bytes)	New frame loss probability
-87	1105	0,0058
-84	1079	0,0058
-79	1479	0,0058

Table A.2: New MTU for frame loss probability minimization

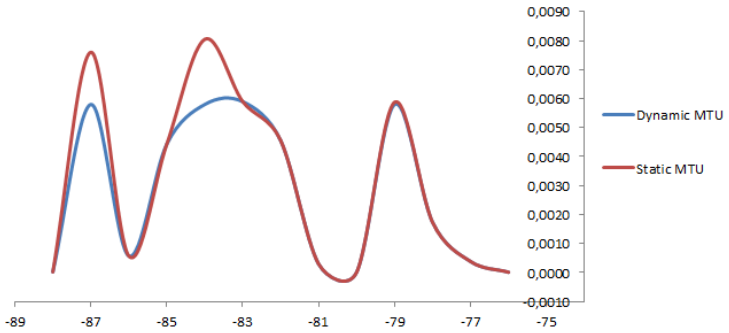

Figure A.2: Frame Loss Probability vs signal (dBm) for adaptive modulation

A.2 Experimental Results

The results presented in this section only refer to that version and static MTU. Only lower RSSI iterations are presented due to the adaptive modulation which changes modulations from mcs0 to mcs7 in order to maintain the communication stable.

A.2.1 TCP Throughput

Table A.3 presents the TCP throughput according to RSSI. When compared to tests from Section 4.7.1, higher limits were achieved due to the use of higher bitrates.

By analyzing Table A.4, it can be seen that there are no relevant gains from dynamic MTU. As concluded in Chapter 4, using adaptive modulation does not bring any improvements. Knowing that our algorithm changes MTU at RSSI values -87, -84 and -79 dBm, data from this table shows

	Static MTU				Dynamic MTU v1			
RSSI	**N**	**Mean**	**StDev**	**Median**	**N**	**Mean**	**StDev**	**Median**
-87	62	8,78	3,50	9,64	237	8,12	3,67	8,13
-86	105	9,95	2,40	10,60	369	10,24	2,29	11,10
-85	198	11,10	2,17	11,20	546	11,09	1,95	11,40
-84	352	12,38	2,95	12,00	659	12,31	2,66	11,90
-83	741	15,31	4,00	14,40	833	15,37	3,90	14,90
-82	1407	18,19	3,86	19,20	1011	18,80	3,92	19,60
-81	1460	20,31	2,84	21,20	1024	21,91	2,90	22,90
-80	757	21,72	2,30	22,20	1040	23,52	2,45	23,60
-79	418	22,98	2,41	22,90	1080	26,16	3,15	25,65
-78	219	25,02	3,69	24,70	1320	29,61	2,92	30,30
-77	79	27,63	4,07	29,00	1171	31,50	2,13	31,60

Table A.3: Throughput (Mbit/s) for Dynamic and Static MTU v1 using TCP

Gains from Dynamic MTU regarding Static MTU (throughput)			
RSSI	**Mean Gain (%)**	**StDev Gain (%)**	**Median Gain (%)**
-87	-7,50	4,91	-15,62
-86	2,93	-4,75	4,72
-85	-0,10	-10,18	1,79
-84	-0,56	-9,96	-0,83
-83	0,37	-2,60	3,47
-82	3,39	1,48	2,08
-81	7,91	2,36	8,02
-80	8,28	6,26	6,31
-79	13,85	30,81	12,01
-78	18,34	-20,94	22,67
-77	14,01	-47,66	8,97

Table A.4: Comparison between Dynamic and Static MTU systems (Throughput TCP)

mean gains of -7,5%, -0,56% and 13,85% respectively. Negative gains are small and the positive gain at -79 dBm is not significant since at higher RSSI, the gains are similar. This can be explained by the fact that the tests were performed in different conditions, leading to higher interference in the static MTU. Also, the fact that the MTU is never changed for values below 1000 bytes can be seen as not very effective.

The plots from Figure A.3 shows the throughput variations for static and dynamic MTU in a different way. Please note that the curves are identical showing no relevant improvements. Also, when compared to plot from Figure 4.12 it can be seen that throughput variation is exponential when using one modulation, being smoothed and with a constant grown when adaptive modulation mechanism is in use.

A.2.2 Round-Trip-Time

RTT tests performed with adaptive modulation are presented in Table A.5.

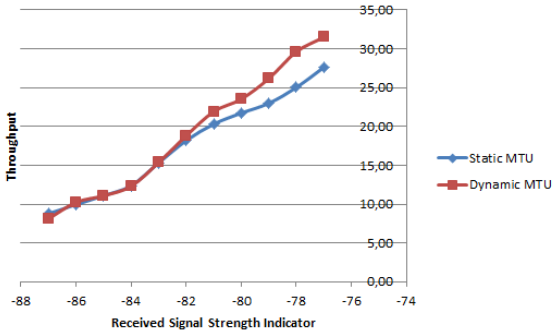

Figure A.3: TCP mean Throughput (Mbit/s) according to RSSI (dBm)

Method	**N**	**Mean**	**StDev**	**Median**
Static MTU	255	13,73	69,94	1,33
Dynamic MTU	255	6,49	46,17	1,23
Gain (%)	-	-52,71	-33,99	-7,43

Table A.5: Ping RTT (ms)

These results as the ones from Table 4.17 show a gain in mean RTT when using dynamic algorithm. As shown in Section 4.7.4, these results cannot be considered conclusive due to the median gain of about 7%.

A.2.3 Conclusion

With the results presented in this appendix it was meant to justify the statements from chapter 4 and so the reason why modulation was fixed. Thus, it was shown that, by using adaptive modulation, it was hard to know the bitrate in use, leading to wrong frame loss probability estimation by our algorithm. Consequently, wrong MTU change decisions are taken, leading to negligible gains. On the other hand, mistakes from these experiments such as not saturating channel and only using MTU reductions above 1000 bytes were seen as very useful contributions in order to accomplish the implementation and the experiments for fixed modulation.

References

[1] Alix3d3 board. http://pcengines.ch/alix3d3.htm/. [Online; accessed 28-January-2014].

[2] Dee Denteneer, Xavier Pérez Costa, and N E C Laboratories Europe. The IEEE 802.11 Universe. (January):62–70, 2010.

[3] Miguel Erazo and Yi Qian. SEA-MAC: A Simple Energy Aware MAC Protocol for Wireless Sensor Networks for Environmental Monitoring Applications. *2007 2nd International Symposium on Wireless Pervasive Computing*, pages 258–262, 2007.

[4] Echanisms For. IEEE 802 .11n MAC frame aggregation maechanisms for next generation high throughput WLANs. (February):40–47, 2008.

[5] R. G. Garroppo, S. Giordano, and D. Iacono. Experimental and Simulation Study of a WiMAX System in the Sea Port Scenario. *2009 IEEE International Conference on Communications*, pages 1–5, June 2009.

[6] J Joe, S K Hazra, S H Toh, W M Tan, and J Shankar. 5.8 GHz Fixed WiMAX Performance in A Sea Port Environment. *Vehicular Technology Conference, 2007*, pages 879–883, 2007.

[7] Masood Khosroshahy. Study and Implementation of IEEE 802.11 Physical Channel Model in YANS (NS3 prototype) Network Simulator Table of Contents. (November), 2006.

[8] David Kidston. Challenges and Opportunities in Managing Maritime Networks. *2008 IEEE Communication Opportunities*, (October):162–168, 2008.

[9] Michael N. Krishnan, Ehsan Haghani, and Avideh Zakhor. Packet Length Adaptation in WLANs with Hidden Nodes and Time-Varying Channels. *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–6, 2011.

[10] Silva Santos Luciano. Wi-Fi Maritime Communications using TV White Spaces. *MSc Thesis. Faculdade de Engenharia da Universidade do Porto*, July 2013.

[11] LAN Man, Standards Committee, and Ieee Computer. *Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, volume 2012. 2012.

[12] Matos Lopes Mário. Comunicações marítimas Wi-Fi usando a banda 5.8 GHz. *MSc Thesis. Instituto Superior de Engenharia do Porto*, October 2013.

[13] RouterBOARD R52n-M. http://i.mt.lv/routerboard/files/R52n-M.pdf/. [Online; accessed 28-January-2014].

[14] BU-353 GPS Receiver. http://www.usglobalsat.com/store/download/62/bu353_ds_ug.pdf/. [Online; accessed 28-January-2014].

[15] Alvaro Suárez, Kholoud Atalah Elbatsh, and Elsa Macías. Gradient RSSI Filter and Predictor for Wireless Network Algorithms and Protocols. *Network Protocols and Algorithms*, 2(2):1–26, June 2010.

[16] Hanwu Wang, Weijia Jia, and Geyong Min. Effective Channel Exploitation in IEEE 802.16j Networks for Maritime Communications. *2011 31st International Conference on Distributed Computing Systems*, pages 162–171, June 2011.